

Tailoring Dependency Models to NLP Tasks

Benjamin Kolz

TESI DOCTORAL UPF / 2016

DIRECTORS DE LA TESI

Dr. Toni Badia

Dra. Roser Saurí

DEPARTAMENT DE TRADUCCIÓ I CIÈNCIES DEL LLENGUATGE



Meiner *Familie* gewidmet.

Dedicado a mi *familia*.

Acknowledgements

First of all I would like to thank my supervisors Dr. Toni Badia and Dra. Roser Saurí for the great collaboration over the past years.

Roser, there have passed a bit more than 5 years since we've worked together for the first time. This was still in Barcelona Media, where I was an intern for two month. This internship turned out to be a collaboration for much more time in the end and the result was our tool for the extraction of time expressions. I want to thank you for helping me to find this way and for all the meetings and ideas you've brought up during the last years!

Toni, thanks a lot for the help over the last years. I remember that some years ago I had classes with you as professor (that was Automatic Translation), afterwards I had the chance to work with you at Barcelona Media and, finally, over the last years you were the supervisor of this dissertation. I want to thank you for all the meetings we had, for all the ideas you put in, and for letting me take part of your knowledge. It has always been a pleasure to work with you!

I would also like to thank my GLiCom colleagues and the UPF professors and lecturers with whom I had the possibility to work.

Especially, I would like to thank Carme Colominas, who was my Catalan teacher many years ago, my Master's supervisor some years ago, and my colleague over the last years. And I have to say that in all three roles it was a real pleasure to work with you!

A big thanks also to Juanma Garrido. Our collaboration within the EmotionFinder was a great experience. The project was a good way to see a different area of NLP (besides dependencies) and the seminars I gave within your university course were fun. I appreciate the opportunity to have worked with you over the last years.

I also want to mention here my colleague Núria Bel, with whom I've been giving classes in two different seminars at the UPF over the last three years. I have to say that it was always a pleasure and that's mainly due to your great organisation, good humour and interest in solving the slightest doubt I had. Thanks a lot!

A thank you also to my office colleagues who know the best how the last years as PhD student felt. Greetings to: Elena, Ioana, Laura, Christian, Rebecca, Eugenio, Montse, María and Alba.

I would also like to mention two of my former professors who have been important for me to find my academic way: Imma Martí and Jaime da Silva, thank you.

Having thanked now many people who worked with me at the university, I want to come now to those people who were there for me outside of the university.

First of all, my wife Diana and my daughter Valentina. Diana, you have been my biggest supporter throughout the last years. We have worked together as a team more and more, and the finishing of this doctoral thesis is also due to your encouragement, your help and your love. Valentina, your birth will not only count towards the most important events of your life but also of mine. Since you are with us, I have a new motivation in life and a new daily inspiration. Honestly, it's incredible how fast you learn and how much happiness you spread wherever you go. Thank you both for being in my life, I love you.

I would like to thank my family in Germany and in Peru. My parents Christian and Petra for supporting us throughout the last years and encouraging me to tackle this dissertation. Oksana, Michaela, Miriam and Julia for being the best sisters one can wish. My grandmothers Helga and Gertrud for caring about us so much even if we are at distance, my grandfather Günther for being an inspiration with his interest for technology (he's 90 years old) and my other grandfather Günter for guiding us still from far away. My parents-in-law Tomás and Shery for their support over all these years, and my brother-in-law Christian for being there for us everytime we visit Peru.

I want to say thank you also to our friends.

To our German friends for keeping the contact and giving us the feeling that we never left Langenberg: Michael, Dandan, Max, Julia, Julian, Lea, David, Jan, Michelle, Thomas, Denise, Uwe, Lucky, Eva, Michael, David und Andrea.

To our friends in Spain for letting us feel at home here: Karla, Pedro, Laia, Carlita, Nelo, Valentina, Cinthya, Brenda, Rodolfo, Fabiola, Carlos, Diasnina, Jordi, Cristina, Pauleta, Blas, Marta, Fabiola and Sam.

To our friends in Peru for being there for us everytime we need them: Luchín, Mayra, Andrea, Jozami, Sugei, Hizami, Elsitá, Anita, Cesar, Lucho, Jorge and Brenda.

To our friends in Ireland: Pau and Lourraine, thank you very much for your friendship and also for the proofreading!

To our friends from France: Kevin and Christelle, merci!

Abstract

Currently available dependency structures differ significantly in the linguistic criteria they are based on, but are not always adequate for their later use in natural language processing tasks. This dissertation analyses the needs of some of these tasks, in particular temporal and discourse parsing, and suggests task-based dependency structures. A surface-syntax dependency structure is taken as base version, which is then tailored to the needs of the corresponding task by means of head selection, customised syntactic function tagset and collapsed dependencies. The work is grounded on the Spanish corpus AnCora, establishing a surface-syntax base version from its constituent structure level. Two dependency models are created, Temporal and Discourse Dependencies, which take the base version as input and adapt it automatically to the task-based versions. The resulting versions are evaluated by network analysis methods, which confirm the adequacy of these new dependency structures with respect to the specific tasks.

Resumen

Las estructuras de dependencias disponibles actualmente incluyen diferencias significativas en cuanto a los criterios lingüísticos en que se basan, y no siempre son adecuadas para su uso en tareas del procesamiento de lenguaje natural. Esta tesis doctoral analiza las necesidades de algunas de estas tareas, concretamente los análisis temporal y discursivo, y propone la creación de estructuras de dependencias orientadas a las mismas. Una estructura de dependencias puramente sintáctica sirve como versión básica que se adapta a las necesidades de cada tarea a través de la selección del head, de un etiquetario de funciones sintácticas adecuado y del recorte de algunos caminos de dependencias. El proyecto parte de la versión castellana del corpus AnCora y establece una versión puramente sintáctica a partir de sus estructuras de constituyentes. Se crean dos modelos de dependencias, Temporal and Discourse Dependencies, que usan la versión básica como entrada y la adaptan de forma automática a las versiones orientadas en las tareas específicas. Las versiones resultantes se evalúan a través de métodos de análisis de redes, que confirman la adecuación de estas nuevas estructuras de dependencias para las respectivas tareas.

CONTENTS

Abstract	ix
List of Figures	xvii
List of Tables	xviii
1. Introduction	1
1.1 Objectives of this Dissertation	4
1.2 Hypothesis	5
1.3 Structure of the Thesis	6
2. State of the Art	9
2.1 Dependency Parsing	9
2.1.1 Dependency Grammar	10
2.1.2 Dependency Parsing Approaches	13
2.1.2.1 Machine Learning Approaches	13
2.1.2.2 Grammar Based Approaches.....	15
2.1.3 Dependency Representations	16
2.1.4 Usage in Computational Linguistics	18
2.2 Temporal Parsing	19
2.2.1 Temporal Parsing Tasks	19
2.2.1.1 Time Expression Recognition	19
2.2.1.2 Normalisation	20
2.2.1.3 Temporal Relation Identification	21
2.2.2 Temporal Information Annotation Schemes	22
2.2.2.1 MUC-TIMEX	22
2.2.2.2 TIDES	23
2.2.2.3 STAG	24
2.2.2.4 Time ML	25
2.2.3 Main Approaches to Temporal Parsing	27
2.3 Discourse Parsing	28
2.3.1 Discourse Parsing Tasks	28

2.3.1.1	Discourse Segmentation	29
2.3.1.2	Relations among Discourse Segments	29
2.3.2	Discourse Representations	30
2.3.2.1	At Textual Level	30
2.3.2.2	In Logics	39
2.3.3	Discourse Parser	43
3.	Methodology	47
3.1	Corpus	47
3.1.1	AnCora Corpus Description	48
3.1.1.1	Morphological Information	49
3.1.1.2	Syntactic Information	50
3.1.1.3	Semantic Level	50
3.1.2	Dependency Annotation	50
3.1.3	AnCora Dependencies Issues	54
3.2	Dependency Customisation	55
3.2.1	Main Ideas	56
3.2.2	Evaluation Approach	59
3.3	Network-based Evaluation	60
3.3.1	Introduction	61
3.3.2	Research on Language Networks	62
3.3.3	Network Concepts	65
3.3.4	Network Measures	67
3.3.4.1	General Network Measures	67
3.3.4.2	Node-specific Measures	68
4.	Surface Syntax Dependencies	75
4.1	Motivation	75
4.2	Related Work	76
4.3	Annotation Description	77
4.3.1	Linguistic Criteria	77
4.3.2	Automatic Conversion	86

4.3.3 Evaluation	89
4.4 Multiword Deconstruction	92
4.4.1 Multiword Theory and AnCora Treatment	93
4.4.1.1 Multiword Definition	93
4.4.1.2 Internal Structure of Multiwords	93
4.4.1.3 External Relations	94
4.4.2 Multiword Deconstruction Process.....	95
4.4.2.1 Motivation	96
4.4.2.2 Multiwords Statistics	96
4.4.2.3 Algorithm	97
4.4.3 Evaluation of Multiword Deconstruction	99
4.4.4 Final Version of Ancora Surface Syntax Dependencies	100
5. Temporal Dependencies	103
5.1 Ways to Express Time in Spanish	103
5.1.1 Lexical Markers	104
5.1.1.1 Anchoring Time Expressions	104
5.1.1.2 Ordering Lexical Markers	107
5.1.2 Grammar	108
5.1.2.1 Tense	108
5.1.2.2 Sequence	112
5.1.2.3 Modality	113
5.1.2.4 Aspect	114
5.1.2.5 Mood	115
5.2 Dependency Relations Customised to Temporal Parsing	117
5.2.1 Approach	118
5.2.1.1 Addition of Temporal Information	118
5.2.1.2 Linguistic Decisions for Head Selection	119
5.2.2 Temporal Dependencies Tagset	130
5.2.3 Implementation	132
5.2.3.1 Algorithm	133
5.2.3.2 ENTimex-Recognizer	134

5.2.3.3 Rules	135
5.2.4 Customisation Results	136
5.2.4.1 Evaluation Corpus	136
5.2.4.2 Result Data	137
5.2.4.3 Error Analysis	138
5.2.4.4 Conclusion	140
6. Discourse Dependencies	141
6.1 Linguistic Approach to Discourse Parsing	142
6.1.1 Discourse Construction in Spanish	142
6.1.2 Discourse Dependencies Relations	145
6.1.2.1 Introduction	145
6.1.2.2 Relation Set	146
6.2 Discourse Parsing Customised Dependency Creation	157
6.2.1 Approach	157
6.2.2 Linguistic Criteria	158
6.2.2.1 Coordinations	158
6.2.2.2 Complementisers	161
6.2.3 Processing-related Assumptions	161
6.2.3.1 Several Discourse Relations at One Head Node	161
6.2.3.2 Disconnection of Nodes in the Annotation	162
6.2.4 Tagset for Discourse Dependencies	164
6.2.5 Implementation	166
6.2.5.1 Algorithm	166
6.2.5.2 Lexical Markers and Linguistic Patterns	167
6.2.6 Customisation Results	175
6.2.6.1 Evaluation Description	176
6.2.6.2 Evaluation Corpus	177
6.2.6.3 Result Data	177
6.2.6.4 Error Analysis	178
6.2.6.5 Conclusion.....	183

7. Evaluation	185
7.1 Motivation	185
7.2 Evaluation Description	186
7.2.1 Hypothesis	186
7.2.2 Network Construction	187
7.2.3 List of Relevant Nodes for Evaluation	188
7.2.4 Measures.....	189
7.3 Results	190
7.3.1 General Observations	190
7.3.2 Temporal Model	191
7.3.3 Discourse Model	198
7.4 Conclusion	200
8. Conclusion	203
A. Confusion Matrix for Functions in SSD Evaluation Corpus	211
Bibliography	213

List of Figures

Figure 1: RST diagram for <i>evidence</i> example text	38
Figure 2: DRS for example 1	40
Figure 3: DRS for example 2	40
Figure 4: Constituent syntax diagram	41
Figure 5: Morphological information example	49
Figure 6: AnCora dependency annotation	51
Figure 7: Example for a syntactic dependency network	70
Figure 8: Example for network neighbours in a syntactic dependency network	71
Figure 9: Example for betweenness in a syntactic dependency network	74
Figure 10: Conversion Algorithm	87
Figure 11: Evaluation corpus fragment	90
Figure 12: Multiword part-of-speech examples	93
Figure 13: AnCora multiword example.....	94
Figure 14: Example for upper head, internal head and lower dependent	95
Figure 15: Example of multiwords and their variety of dependents	95
Figure 16: Multiword deconstruction algorithm	97
Figure 17: Multiword classifier output	98
Figure 18: Example of a deconstructed multiword with a coordination inside	99
Figure 19: Empty sentence in AnCora constituents	101
Figure 20: Empty sentence in AnCora dependencies	102
Figure 21: SSD dependencies format example	126
Figure 22: Temporal Dependency Annotation algorithm	133
Figure 23: Recognizer performance	135
Figure 24: Discourse Dependencies algorithm	166
Figure 25: SSD corpus example for network construction	187
Figure 26: Incoming connections for <i>vigilar</i>	193
Figure 27: Outgoing connections for <i>vigilar</i>	194
Figure 28: Total-Degree connections for <i>vigilar</i>	195
Figure 29: Betweenness graph example	196
Figure 30: Node Neighbours in graph example	196

List of Tables

Table 1: RST Relations	37
Table 2: AnCora corpus size overview	48
Table 3: AnCora dependency function tagset	53
Table 4: Edge list without weights	65
Table 5: Edge list with weights	65
Table 6: Lemma edge list	66
Table 7: Lemma edge list with weights	66
Table 8: Not considered Stanford tags	83
Table 9: Renamed Stanford tags	84
Table 10: Tags added from AnCora dependencies tagset	84
Table 11: Tag added to SSD dependencies tagset	84
Table 12: Surface-Syntax Dependencies function tagset	85
Table 13: Evaluation corpus size overview	89
Table 14: SSD Conversion Result	90
Table 15: Example of inconsistencies in AnCora multiword treatment	96
Table 16: Multiword lengths statistics	97
Table 17: Multiword deconstruction evaluation	100
Table 18: SSD tagset sorted by frequency	100
Table 19: Markers for temporal clauses	124
Table 20: Temporal Dependencies tagset	130
Table 21: Tags without usage in Temporal Dependencies annotation	132
Table 22: General adaptation results for Temporal Dependencies	136
Table 23: Erroneously created tags	139
Table 24: Discourse relations overview	146
Table 25: Discourse Dependencies Tagset	164
Table 26: Tags without usage in AnCora Discourse Dependencies	165
Table 27: Prepositions as lexical marker for attribution.....	167
Table 28: Verbs as lexical markers for attribution	167
Table 29: Adverbial clauses introducing conjunctions for enumeration	169
Table 30: Prepositions introducing infinitival clauses for enumeration	169
Table 31: Adverbial modifiers for enumeration	170

Table 32: Lexical markers for inclusion	170
Table 33: Patterns for inclusion	171
Table 34: Lexical markers for opposition	172
Table 35: Conjunctions for causality (conditions)	173
Table 36: Conjunctions for causality (cause)	173
Table 37: Conjunctions of causality (effect)	173
Table 38: Conjunctions of causality (purpose)	174
Table 39: Prepositional structures for causality (purpose)	174
Table 40: Prepositional structures for causality (cause)	175
Table 41: Discourse relation identification results	177
Table 42: General adaptation results	178
Table 43: General network measures	189
Table 44: Node-specific network measures	190
Table 45: Result comparison between SSD and Temporal Dependencies	192
Table 46: General Network characteristics for SSD and Temporal Dependencies.....	197
Table 47: Result comparison between SSD and Discourse Dependencies	199
Table 48: General Network characteristics for SSD and Discourse Dependencies.....	200

List of original publications

Chapter 4:

Kolz, B., Badia, T. and Saurí, R. (2014). From constituents to syntax-oriented dependencies. *Procesamiento del Lenguaje Natural*, 52, pp. 53-60.

1. Introduction

Since the beginning of natural language processing (NLP), syntactic analysis is central in obtaining the relevant linguistic information from texts. Dependency structures are one way to present this information, a way which has been gaining more and more importance for NLP applications in the last years (Sagae, 2009; Yoshida et al., 2014; Liu et al., 2015; Ma et al., 2015).

Their compact representation, while still being informative on syntactic features, made them a good choice for parsing and for machine learning related tasks (e.g. Čmejrek, Cuřín and Havelka, 2003). This is also an advantage in comparison to other representation forms like constituent structures. Furthermore, the relations at word level in dependency structures make it easier to extract semantic relations from the text.

However there is not a unique way in which the linguistic information can be encoded in dependency structures: both the head selection and the labels used for tagging the syntactic relations between two nodes can be designed in different ways. Ongoing research (de Marneffe and Manning, 2008; Silveira and Manning, 2015) on such structures shows that constant improvements are being made, and the variety of used structures (Civit et al. 2006; Mille et al., 2009; Arias et al., 2014) is an indicator that a standard in presentation has not yet been established.

The variety in dependency structures can be seen in the individual annotation criteria regarding the head selection between head-dependent pairs, the applied tagset (in both quantity and information graininess) and the use of collapsed dependencies.

There is, for example, no doubt that a dependency relation exists within the elements of a composed verb form like *ha hablado* ('he has talked'). But there is discussion among linguists concerning whether the auxiliary verb *ha* ('has') depends on the participle *hablado* ('talked') or the other way around (Kolz et al., 2014a). The applied tagset for the dependency annotation is another point where very differing decisions can be taken. It would theoretically be possible to use just one tag, for example *dep*, to mark in the annotation the relation of the dependent to the head. Nevertheless, this would be, at a linguistic level, a very uninformative annotation. The other extreme would be to set up a

specific tag for every dependency relation and as fine-grained as possible. In this case, the detail of the annotation will lead to problems in their further usage. For example, a machine learning approach will have difficulties in the identification of the relevant patterns.

The idea of collapsed dependencies (de Marneffe and Manning, 2008) refers to the possibility to disconnect words from the dependency tree in order to shorten distances between especially relevant nodes for the annotation purpose.

Stanford University has recently done important steps (de Marneffe et al. 2013 and 2014) towards a standard dependency annotation for different languages, but it has still to be proved if a standard format is an advantage for all type of usages or if it is just a base for further investigation. Maybe the variety of NLP tools that make use of dependency structures makes it not desirable to have a standard format due to the differences in linguistic information they require.

A discourse parser (Sagae, 2009; Yoshida et al., 2014), for example, finds important information in conjunctions which connect different discourse segments, while a temporal parser (Kolomiyets and Moens, 2013) heavily relies on temporal expressions and other time-specific lexical markers. If one considers the sentence in example (1), a temporal parser can identify different components of the sentence as containing the relevant information as compared to the point of view of a discourse parser. In (1b) basically the main clause is marked in bold since it describes what happens (*viaja*, ‘travels’) and when this happens (*en tres semanas*, ‘in three weeks’). Both elements are basic information for a temporal parser. The bold in (1c) marks the elements that are especially interesting for a discourse parser. What happens is described in the main clause (*viaja*, ‘travels’) and why this happens is explained by the subordinate clause (*ha ganado*, ‘has won’). The chosen conjunction (*porque*, ‘because’) states explicitly the relation between main and subordinate clauses.

- (1) a. Valentina *viaja en tres semanas porque ha ganado el premio.*
Valentina travels in three weeks because she has won the prize.
b. Valentina **viaja en tres semanas** porque ha ganado el premio.
c. Valentina **viaja en tres semanas porque ha ganado** el premio.

So far, dependency structures have not been adapted to their later use in NLP tasks, but NLP tools have tried to benefit as much as they can from available dependency annotations.

If both a temporal and a discourse parser rely then on the same dependency structures as input data, those structures cannot encode at the same time all the important information for both tools, since dependency structures imply a strict head-dependent relation within the words. If one considers the example of the subordinate clause, it is a hard decision to choose between the conjunction, which introduces this clause, as head of the structure, in order to facilitate discourse parsing, or to have the verb inside the clause as head to make information extraction easier. Both words cannot be the head of the subordinate clause. Furthermore, it has to be taken into account that these two NLP tasks are just an example out of a great variety that exists nowadays.

Nevertheless, all tools, whatever NLP task they perform, have in common that they need quality input data in order to be able to produce quality output. If the tool is rule-based, those rules will have a hard time dealing with errors or inconsistency in the input data. If the tool uses a machine learning approach, the quality of the input data will have an impact on the performance a certain algorithm can reach.

“However, it is not enough to simply provide a computer with a large amount of data and expect it to learn to speak—the data has to be prepared in such a way that the computer can more easily find patterns and inferences. This is usually done by adding relevant metadata to a dataset. Any metadata tag used to mark up elements of the dataset is called an annotation over the input. However, in order for the algorithms to learn efficiently and effectively, the annotation done on the data must be accurate, and relevant to the task the machine is being asked to perform. For this reason, the discipline of language annotation is a critical link in developing intelligent human language technologies.”

(Pustejovsky and Stubbs, 2012:2)

The motivation for this dissertation is therefore to show how different models of dependency structures adapt in an optimal way to their further usage in NLP applications and how these models can be created automatically from a standard annotation structure.

1.1 Objectives of this Dissertation

The main goal of this doctoral thesis is to analyse linguistically the current situation of dependency structures and investigate possible solutions for the optimisation of those structures according to their later use in NLP tasks.

The first objective has to be the study of the linguistic needs of the chosen NLP tasks. For this project temporal and discourse parsing were chosen as such tasks, but the list could theoretically be expanded far beyond that, since the NLP research field gathers a wide variety of different tasks.

The linguistic observations have to be converted into criteria which serve for the task-based customisation of dependency structures. This is why it is important also to study current annotation schemes which are used nowadays for dependency annotation and for the chosen NLP tasks in order to design dependency models which adapt in the best possible way to the future use of the data. The task-oriented models shall on the one hand make it easy to extract relevant information and, on the other, enhance the given data by task-specific information.

This theoretical objective translates into two further practical objectives, which are the creation of a temporal-oriented and a discourse-oriented dependency model. These models shall include the identified linguistic criteria, facilitate the extraction of relevant information and be enhanced by temporal and discourse information according to the task of each model. The whole adaptation process will be implemented by automatic means and prove in this way its feasibility in NLP pipelines. The approach is thus to take an already dependency annotated corpus and to automatically create two task-based dependency annotations for specialised NLP purposes by means of the constructed models.

The next objective is the evaluation of the implemented task-oriented optimisation process. While the adaptation from a base dependency corpus into the task-based dependency annotations includes an evaluation of the conversion process, a final evaluation still has to substantiate with empirical data the improved dependency structures.

A secondary objective of this work is to create linguistic resources for the research community. The dependency-annotated corpora will be available for further research and applied purposes as NLP parsing. This will contribute to the progress in the field of dependency research, and will help to improve parsing results.

1.2 Hypothesis

The creation of two task-oriented dependency models will show how different dependency annotations can be constructed depending on their goal and which linguistic decisions are implied in each approach. The linguistic criteria can be introduced into a dependency annotation by means of the head selection between dependent-head pairs. This will facilitate short paths in the dependency structure between nodes, which are considered important for the specific annotation. The syntactic tagset of the dependency annotation can be customised to enhance the annotations with task-specific information.

The adaptation of a standard dependency corpus to task-specific dependencies should be proved as feasible by automatic means, as automatic conversions between constituent and dependency annotations have already been shown (Civit et al., 2006). The resulting task-based dependency structures should be designed in a way that the linguistic features, that are important for the specific NLP tasks, are well positioned within those structures and easily reachable by NLP tools.

The evaluation of the customised dependency structures can be conducted in two ways. The first consists in experimenting with a temporal or discourse parser using the resulting dependencies in a comparison to not task-oriented dependencies. The second option is the use of task-independent evaluation metrics over the resulting dependency models. A possibility is here to apply network-based analysis methods. In this way the implemented changes can be observed from a more theoretical approach (Ferrer i Cancho et al., 2004). Both options are valid to measure the improvement of the dependency structures. Nevertheless, several points are in favour of the second option. The practical approach by means of a discourse or temporal parser presupposes the free access to such tools and that they accept the available corpora as input data. This is

difficult to guarantee at this moment and the necessary time for the implementation of two dependency models into two different task-specific pipelines would be out of the scope of this dissertation. On the other hand, there are tools for network analysis freely available (Csárdi and Nepusz, 2006), which can be perfectly used for the evaluation of this project. Furthermore, the scientific value of the second option seems to be significantly higher, since the created corpora will be made available for free and the probability of them to find future use in NLP parsing projects is higher than the possibility to see a comparison by means of a network analysis. As a result, the second option seems to be more feasible and with a more valuable approach.

1.3 Structure of the Thesis

In the present chapter, the reader finds the introduction to the investigation proposed for this thesis. It explains the **motivation** for this project and the **objectives** that shall be reached.

Afterwards, Chapter 2 gives an overview of the current **state-of-the-art** of the implied linguistic areas. This includes dependency parsing, as a general topic, and temporal and discourse parsing as NLP tasks chosen for the task-based dependency model creation.

Chapter 3 explains the **methodology** of the project. It gathers information about the used resources, basic ideas about the model creation and the plan for the evaluation of the project.

Chapter 4 presents **Surface Syntax Dependencies** (SSD). This is a dependency model based on purely syntactic criteria, which is applied to an automatic conversion from AnCora constituents into SSD dependencies.

In Chapter 5, the first task-based dependency model is shown, which is called **Temporal Dependencies**. It presents an automatic adaptation from the surface-syntax version to a temporal parsing optimised dependency annotation.

The second task-based model is named **Discourse Dependencies** and described in Chapter 6. It presents a dependency annotation adapted to discourse parsing.

Chapter 7 shows the final **evaluation** of the optimisation process between the surface-syntax model and the two task-based dependency models by means of a network analysis.

Finally, the **conclusion** in Chapter 8 gives further insights into the overall results of this thesis, presents its contributions and proposes future work based on it.

2. State of the Art

Chapter 2 presents the most important state-of-the-art work on linguistic theories related to the project of this thesis. The main topic is dependency parsing, therefore first of all some basic ideas are presented regarding this area (Section 2.1). It includes a look at Dependency Grammar and different ways of dependency parsing itself. Finally, some words are dedicated to state-of-the-art representations and the usage of dependencies in NLP tools.

The next sections present the theoretical frameworks related to the specific areas of which adapted dependency structures will be constructed in this project, namely temporal (Section 2.2) and discourse parsing (Section 2.3). Both individual topics have a wide linguistic background which will be presented here according to its relevance for the creation of task-based dependency structures.

The extensive content regarding state-of-the-art shows on the one hand that there is ongoing research regarding dependency parsing, and on the other hand the need for the consideration of the NLP task-related theories in the creation process of dependency structures.

All upcoming theories and examples refer to the working languages Spanish and English, other languages can differ very much in their way to express this information.

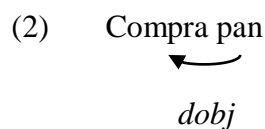
2.1 Dependency Parsing

Dependency parsing is based on the idea of Dependency Grammar which dates back primarily to Tesnière (1959). He proposed already a formalisation of syntactic structures which saw the verb at highest hierarchic level and which distinguished syntax clearly from semantics and morphology. His focus was also the hierarchical syntactic order in a sentence in contrast to textual linearity. Dependency parsing relies on many of his ideas. Dependency relations take the (finite) verb as centre point of all clause structure and all other syntactic units (words and punctuation marks) depend then directly or indirectly on the verb. In contrast to constituency grammars, dependency grammars do not include

phrasal nodes and have found a wide usage in computational linguistics in the last years. The present section is divided into a first part which explains dependency grammar (Subsection 2.1.1), followed by a subsection about dependency parsing itself (Subsection 2.1.2) and finished by a look at its usage in computational linguistics (Subsection 2.1.3).


2.1.1 Dependency Grammar

Modern theories of dependency grammar are usually referred back to the French linguist Lucien Tesnière whose work was published posthumously in 1959. Different theories based on this idea came up since then (e. g.: Mel’čuk, 1988; Hudson, 1984; 1990; Hellwig, 1986, 2003; Starosta, 1988). All of them have in common that they analyse a sentence in terms of dependency relations which establish a binary, asymmetrical relation between syntactic units. The term *syntactic unit* refers to the words and punctuation marks of the analysed sentence. Thus it refers to all tokens in a tokenised sentence. A dependency relation holds between two syntactic units, one is called the *dependent* and the other, on which former depends, is called the *head*. Additionally the *dependency type* indicates the syntactic function of *dependent* to *head* (Kübler et al. 2009:2). In (2) *Compra* (‘Buy’) is head of *pan* (‘bread’), which is the direct object and which therefore shows this in its syntactic function (here: *dobj*).¹



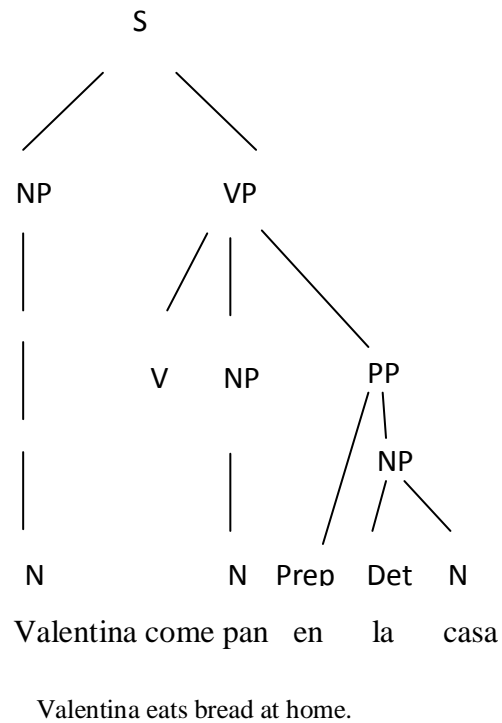
Normally, the head of the whole sentence gets the *dependency type ROOT* as it does not have a head within the sentence. Optionally, the artificial token *ROOT* can also be inserted, as seen in (3), so that all words have a syntactic head. The insertion of a token for *ROOT* gives an advantage in formal definitions and computational implementations.


¹ The arrows point to the head in the examples throughout this dissertation

(3) ROOT Come pan


The most widely used syntactic structures are *phrase structures* which see their origin in Chomsky (1957) and which are different to dependency structures as they group words into phrases (4). These phrases are classified by structural categories as noun phrases or verb phrases. The dependency structure (5) does not represent the phrasal information but shows head-dependent relations between words (to be more concrete, *tokens*) which are classified by functional categories.

(4)



(5) ROOT Valentina come pan en la casa.


Dependency structures follow a hierarchical sentence structure with internal items corresponding to tokens which are present in the natural language text. A hierarchical order is preferred to a linear order in a syntax analysis of a sentence. Phrase structures on the other hand contain abstract items (such as *noun phrase*) and group together tokens into phrases. Text linearity is an important point here as word order in English and especially in Spanish is not always fixed. Dependency structures are able to handle non-projective structures (Mel'čuk and Pertsov 1987; Gerdes and Kahane, 2007). Phrase structures need to treat this topic by complex mechanisms such as transformations (Gerdes, 2006), even if work on discontinued constituents has been done (Becker et al., 1991). In phrase structures, the hierarchy is introduced through the connections of the structural categories.

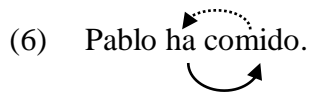
The phrase structure can implicitly be found in the dependency structure as a word and its dependents can build a phrase in one or several steps. Therefore one can say that this information is just not explicitly encoded in dependency structures. On the other hand, functional relations are not explicitly encoded in phrasal structures but can be identified in terms of structural configurations. The noun phrase directly under the highest node S (sentence) could be identified as subject, for example.

An automatic conversion between those two structures is nevertheless a difficult task but has proved to be feasible at least from phrase structures into dependencies (Civit et al., 2006). This is due to the fact that constituent structures are usually more complex and contain more information. Rambow (2010) argues that the conversion quality is not due to the direction of conversion between these two representations but only due to the content given by the source structure. Recent work, such as Kong et al. (2015), has also shown good results in the conversion from dependencies into phrase structures.

As both dependency structures and phrase structure describe syntax, they describe features related to each other and can also be represented together in so called hybrid representations (Wang and Zhang, 2010). It has to be taken into account that they do not necessarily give the same syntactic information. Dependencies usually give functional syntax information, while constituents give structural syntax information.

The dependency structure can also vary according to the linguistic criteria chosen for the annotation (Schwartz, 2012; Silveira, 2015). A syntactic approach shows differences to a semantic approach in various aspects of the structure such as head selection. This can be seen in structures such as composed verb forms in which both auxiliary and main verb could be the head depending on the chosen criteria (6).

verbal object (syntactic approach)



auxiliary verb (semantic approach)

The encoded information dependency representations can therefore vary in a significant way and affect also the syntactic functions used between head and dependent pairs.

2.1.2 Dependency Parsing Approaches

Dependency parsing refers to the task of an automatic dependency structure analysis of a given sentence. Different methods can be applied to reach this goal from machine learning (Nivre et al., 2006) to formal grammar approaches (van der Beek et al., 2002). Data-driven approaches make use of a large set of syntactically annotated sentences, while grammar-based approaches define dependency structures based on the given grammar.

2.1.2.1 Machine Learning Approaches

Machine learning approaches have found considerable use in recent years (Nivre et al., 2006; Duan et al., 2007). They can further be divided in *supervised* and *unsupervised* approaches.

Unsupervised machine learning has the advantage that there is no need for annotated training data, but on the other hand results have been inferior to supervised techniques so far (Klein, 2005).

Supervised machine learning approaches take as input for learning a set of correctly annotated sentences with their dependency structure (Kübler et al., 2009:7). Afterwards two problems have to be solved. On the one hand a *learning problem* as the parser has to learn efficiently from the input set, and, on the other hand, a *parsing problem* as the learned model has to be applied to unseen sentences. Unsupervised approaches do not have access to correctly annotated corpora as input data, which makes their implementation less time consuming but, at the current state of the art, less precise.

The data-driven approaches themselves can vary according to the parsing model, the applied learning algorithm and the chosen parse algorithm. Two classes of data-driven methods are *transition-based* and *graph-based* methods (Kübler et al. 2009:7).

- **Transition-based Parsing**

Transition-based methods make use of a transition system (or state machine) in order to map an unseen sentence to its dependency graph. In this method the learning problem consists of the prediction of the next state transition by help of the transition history. The parsing problem is to apply the induced model for an optimal transition sequence for the input sentence. This method is sometimes also called *shift-reduce dependency parsing* since the approach is similar to deterministic shift-reduce parsing used for context-free grammars. Kudo and Matsumoto (2002) were the first to make use of this technique.

- **Graph-based Parsing**

Graph-based methods define a space of candidate dependency graphs for a sentence. The learning problem is the induction of a model for the assignment of scores to the

candidate dependency graphs for a sentence. The parsing problem is then to find by help of the induced model the highest-scoring dependency graph for the input sentence. This method is also called *maximum spanning tree parsing* because of its similarity in finding the highest-scoring dependency graph. First usage of graph-based parsing can be traced back to Eisner (1996).

2.1.2.2 Grammar Based Approaches

Grammar-based approaches are normally divided into *context-free* (Hays, 1964; Gaifman, 1965) and *constraint-based* (Maruyama, 1990) methods. Grammar-based approaches need generally a formal grammar in order to parse sentences. This grammar can be handcrafted or also be learned automatically from linguistic data. So the grammar-based approach can also include a machine learning component.

- **Context-free**

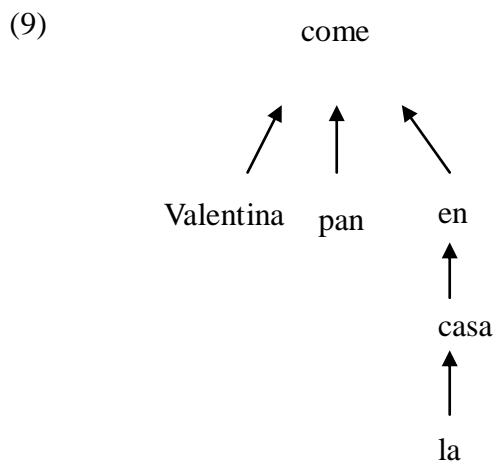
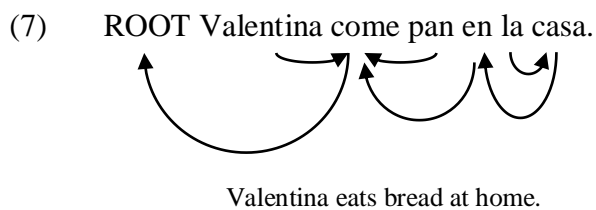
Context-free dependency parsing uses parsing algorithms which were originally developed for context-free grammar. These algorithms include chart parsing algorithms, which find also usage in graph-based parsing, and also shift-reduce type algorithms, which are similar to methods used for transition-based parsing.

- **Constraint-based**

Constraint-based dependency parsing uses a different approach since it sees parsing as a constraint satisfaction problem. The grammar defines a set of constraints which a graph has to fulfil in order to be accepted as a well-formed dependency graph. The parsing problem consists therefore in finding a dependency graph for a sentence that satisfies the constraints given by the grammar.

2.1.3 Dependency Representations

Dependency structures can be represented in different ways. Arrow arcs (7) and brackets (8) can work over the standard sentence structure, while dependency trees convert the sentence into a directed graph (9).



The difference is mainly visual in this case as only the view for the user changes. An NLP program reads the data as a codified list in a certain format and makes no distinction between these ways of representation. A more important way of structure change can be found in *collapsed dependency structures*.

In the latter case, not all representations include all tokens of the sentence. In a non-collapsed dependency representation each token of a sentence is represented by a node

in the tree. In collapsed dependency structures it is possible to remove certain tokens from the dependency representation in case they are considered to be not relevant for later use. This omission normally affects tokens which do not convey semantic information, e.g. certain prepositions. The removed token can be added to the label of the collapsed dependency, so that a parser will still have access to it if necessary. An example of this treatment can be seen in *Stanford Collapsed Dependencies* (de Marneffe and Manning, 2008).

While dependency structures came up having mainly a syntactic focus due to its origins in *Dependency Grammar* (primarily Tesnière, 1959), other frameworks such as *Meaning-text Theory* (Mel'čuk, 1981 and 1988) brought up different points of view. Four different points of view can be seen observed: syntactic, semantic, morphological and prosodic. Most work makes use of a syntactic or semantic focus. The morphological and prosodic approaches only show advantages when working at word level such as in typological studies or in the treatment of clitics. Note that the chosen focus influences directly the head selection in dependency structures.

Works with semantic focus have gained interest for several NLP tools as it facilitates information extraction, while a syntactic point of view is more accurate when an NLP tool relies on linguistic analyses.

In recent years, works at Stanford University have aimed at a standard format for dependency annotations, first for English (de Marneffe and Manning, 2008), and afterwards also across languages in *Universal Dependencies* (de Marneffe et al., 2014). Furthermore, a manually annotated gold standard dependency corpus in English was created for further investigation (Silveira et al., 2014). This is an ongoing investigation line, and the next years will most probably show new aspects. For the moment it shows that, even for English, resources are limited and especially in terms of high quality dependency annotations. In the next years, it will have to be proved if a standard format for one, or even more languages, is the best choice for the different uses the data can have. NLP tools significantly differ from the type of information which is important for them in linguistic data. It is therefore an ambitious plan to aim for a standard annotation scheme for all purposes. Nevertheless, the creation of standard dependency data and further automatic adaptation to the desired NLP tasks seems a feasible solution. The

next section will show some examples for the usage of dependency structures in such tools.

2.1.4 Usage in Computational Linguistics

Many applications in natural language processing rely on syntactic information as their input. While this information can be packed into phrase structures, research in the last years has shown that dependency structures seem to be preferable for many tasks as they facilitate parsing and machine learning purposes. Being compact, they still offer a great part of encoded syntactic information. Examples of usage in different areas can be found for information extraction (Culotta and Sorensen, 2004), machine translation (Ding and Palmer, 2004; Quirk et al. 2005), textual entailment (Haghighi et al., 2005), lexical ontology induction (Snow et al., 2005), question answering (Wang et al., 2007) and grammar checker (Mozgovoy, 2011). Further research on dependency structures will therefore benefit a wide variety of NLP applications. It is remarkable that, in spite of their variety, all NLP applications make use of similar dependency structures as input data. The range of available resources is limited, there are some corpora available for English such as English Web Treebank (Silveira, 2014), Prague English Dependency Treebank 2.0 (Hajič et al., 2012) and BioInfer (Pyysalo et al., 2007), but, considering that English is the language with most use in NLP projects, this is not much. Other corpora, such as Penn Treebank (Taylor et al., 2000), offer only a constituent layer as syntactic information. The situation for Spanish is also very similar with only AnCora (Taulé et al., 2008) and IULA Spanish LSP Treebank (Arias et al., 2014) available with a dependency layer. UAM Spanish Treebank (Moreno et al., 2000) is also an available resource but smaller and only with a constituent layer. These circumstances make it difficult to imagine that NLP tools can work with adequate input data according to their uses. The sparse available variety of dependency corpora makes the tools adapt to what they can get as input data. Sections 2.2 and 2.3 will show how different the needs of linguistic information can be according to the NLP task.

2.2 Temporal Parsing

After having discussed *dependency parsing*, this section explains now the state of the art of the first task for which an adapted dependency model shall be created. This is important as the model shall be adapted to the needs of temporal parsers and encode rich linguistic information as future input for them.

Generally speaking, *temporal parsing* refers to the automatic extraction of time related information in natural language text. Additionally, this can imply the processing of the extracted data as further step as in Llorens et al. (2011). Temporal parsing plays an important role in NLP, as with time information it is possible to order different events in a text with respect to each other and to a timeline.

This section presents first the involved subtasks in temporal parsing (Subsection 2.2.1), then gives an overview of the different annotation schemes for time information (Subsection 2.2.2) and presents state-of-the-art parsing approaches (Subsection 2.2.3).

2.2.1 Temporal Parsing Tasks

This subsection describes shortly the different tasks which have to be carried out for temporal information processing.

2.2.1.1 Time Expression Recognition

The time expression (also: timex) recognition task concerns the exact detection of a temporal expression in natural language text. Time expressions can be seen as those constructions referring to points or intervals on the timeline (Saurí, 2010:3).

This task implies the detection of the starting and end point of the time expression. Additionally, the determination of the type of time expression can be part of this task.

Time expressions can be expressed by different parts of speech such as temporal adverbials (ex.: *weekly*) or nouns referring to weekdays, holidays (ex.: *Thanksgiving Day*) etc.

(10) **El lunes** encontraron oro en México.

On Monday, they found gold in Mexico.

In (10), *El lunes* ('Monday') is marked as temporal expression. The type would have to be selected according to the annotation scheme; in TimeML it would be DATE (see Subsection 2.2.2). Note that time expressions can also be found in phrase structures, such as nominal (*el lunes pasado*, 'last Monday') and prepositional phrases (*en tres semanas*, in three weeks).

2.2.1.2 Normalisation

The normalisation of a temporal expression implies the conversion of it into a standardised format which a computer can handle afterwards. This is an important process as character strings in a certain language without further processing do not mean anything for a machine. Normalised values make it possible for the computer to use this data for further processing. In this normalisation process, it has also to be taken into account that temporal expressions can refer to points on the timeline like specific dates or also to intervals. A standardised format is necessary, as having different possible solutions for the value of a time expression would complicate the further processing by the computer. If one considers the normalisation of a date like *el 3 de noviembre de 2012* (3rd November 2012), it is clear that many solutions can be chosen for its normalised representation. One could for example begin by the day value (integer with range from 01 to 31), take then the month (range 01-12) and at the end the year, what would lead to *03112012*. Another solution would be reversing the order and begin with the year, then the month, at the end the day value and add hyphens after each individual value. The adding of the hyphens would in this case only improve the

readability for human readers, since a computer would have no problem to work without the hyphens.

The important fact for the computer is that the values are standardised and follow certain rules. This is why the use of annotation schemes is important for this process (see Subsection 2.2.2).

- (11) "el 3 de noviembre de 2012", type: date; fully specified: yes normalised value: 2012-11-03 (depending on the standardised format)

The difficulty in the normalisation process is that not all time expressions are fully specified as they appear in the text. The example *el 3 de noviembre de 2012*, as seen in (11), is fully specified, but *mañana* ('tomorrow') could as well be used by the text author and even refer to the same date. However, in contrast to the first example, it would need more information to be normalised. Underspecified time expressions need a time reference, a so called *anchor*, which helps finding the missing information for their normalisation. This can be on the one hand the creation time of the document or, on the other hand, another time expression.

The normalisation task consists therefore in the conversion of an already detected time expression into a standardised format which allows further processing.

2.2.1.3 Temporal Relation Identification

In this task temporal relations have to be found and established. This relation can hold between two events, two temporal expressions or an event and a temporal expression. Pairs of those temporal entities have to be identified and linked. The automatic identification of these relations is not an easy task but an important one as many time expressions and events cannot be ordered on the timeline without knowing to which time point they are related. Expressions like *tres días después* ('3 days later') on its own cannot be set on the timeline, one needs previously their reference point.

2.2.2 Temporal Information Annotation Schemes

This subsection is dedicated to the annotation schemes which are used to represent temporal information in text.

Since 1995, there has been research on standard annotation formats for temporal information in order to facilitate the access to the required data for NLP tools, and to guarantee the quality of comparisons between automatic annotations. Several annotation schemes have come up in the last years, the most important will shortly be introduced in this section. It should be taken into account that they have built up on each other and have implemented improvements in relation to their precursors. The state-of-the-art annotation scheme is therefore the newest of them, which is TimeML. The following subsections explain briefly each annotation scheme and give information about the implemented improvements.

2.2.2.1 MUC-TIMEX

In 1995 started the development of this annotation scheme (Grishman and Sundheim, 1996), called MUC-6, which introduced the TIMEX tag and two types of temporal expressions (DATE and TIME). Only absolute time expressions were annotated, relative time expressions like *last July* were excluded (Mani et al., 2005:489).

Later in 1997 at MUC-7 (Gaizauskas and Wilks, 1998) relative temporal expressions like *yesterday* were added to the scheme. The annotation scheme left still much room for improvement but was already written in a XML based format and used the TIMEX tag to mark temporal expressions in text (12).

One of the main limitations of its tagging guidelines was that context-dependent expressions like *yesterday* were only tagged but not interpreted as full calendrical time (Mani et al., 2005:489).

- (12) "by 9 o'clock Monday"
by <TIMEX TYPE="TIME">9 o'clock</TIMEX><TIMEX
TYPE="DATE">Monday</TIMEX>

2.2.2.2 TIDES (Translingual Information Detection, Extraction and Summarization)

This annotation scheme was developed in the year 2000 (Ferro et al., 2000; Wilson et al., 2001; Ferro et al., 2005) with the purpose of multilingual annotation of temporal expressions. The TIMEX tag was replaced by the TIMEX2 tag and some new attributes were added in order to include more linguistic information within the annotation. An example can be seen in (13).

- (13) "The bombing took place on the second of December."
The bombing took place on <TIMEX2 VAL="1998-12-02">the second
if December</TIMEX2>. (Ferro et al., 2005:13)

Two new types were considered, SET and DURATION, but the TYPE attribute was removed. A new attribute VAL was introduced which represents the normalised value of the temporal expression. This value follows the ISO 8601 standard. Some other attributes were introduced to capture the semantics of timexes.

The MOD attribute captures temporal modifiers, ANCHOR_VAL shows the normalised form of an anchoring DATE/TIME, ANCHOR_DIR contains the relative time direction between VAL and ANCHOR_VAL and SET identifies temporal expressions denoting SETs.

TIDES was used in TERN (2004) evaluation forum and in EVALITA'07. It was the most important annotation scheme (Negri and Marseglia, 2004; Saquete et al., 2006) until TimeML (Pustejovsky et al., 2005) became the standard.

TIDES had the disadvantage that it was limited to the annotation of temporal

expressions. Events and temporal relations were not yet considered. Both are important concepts for temporal reasoning.

2.2.2.3 STAG (Sheffield Temporal Annotation Guidelines)

The creation of STAG was motivated by former research in temporal information processing which showed the need for an extension in temporal information annotation. This extension was created to consider not only timexes but also events and temporal relations in order to improve temporal reasoning capabilities.

STAG was developed by Setzer and Gaizauskas in 2000 (Setzer and Gaizauskas, 2000 and 2001) for the identification of events in news and their anchoring to time as well as their relative ordering. STAG considered three types of tags, which were timex, signal and event. An example annotation is shown in (14).

- (14) A small single-engine plane **<event eid="9" class="OCCURRENCE" tense="past" relatedToTime="5" timeRelType="included" signal="9">** crashed **</event>** into the Atlantic Ocean about eight miles off New Jersey **<signal sid="9">on</signal> <timex tid="5" type="DATE" calDate="12031997">**Wednesday**</timex>**. (Setzer, 2001: 106)

STAG introduced some important points that former schemes did not consider. EVENTS were annotated for the first time as well as temporal signals which show the relation between two temporal entities. It was also possible to annotate temporal relations for the event tag indicating the relation between two events or an event and a temporal expression.

The annotation scheme STAG did not yet include stative events and the temporal relations were part of the EVENT tag, which led to some problems in the application of the scheme.

2.2.2.4 TimeML (Time Markup Language)

The TimeML specification language (Pustejovsky et al., 2003) was developed in the context of the ACQUAINT program on Question Answering Systems and improved many of the shortcomings of its precursors.

It is a rich specification language for the annotation of event and temporal expressions in natural language text. Furthermore, it makes it possible to capture all temporal information and normalise it into a standard format.

It uses four main tags, which are TIMEX3, SIGNAL, EVENT and LINK. The TIMEX3 tag is an advancement of former TIMEX2 tag used in TIDES and encodes all relevant time information about a time expression by means of several attributes (Saurí et al, 2005). The formerly removed *type* attribute was implemented again and has four possible values, which are DATE, TIME, DURATION and SET. Some examples are shown in (15).

- (15) DATE: *el 13 de noviembre* ('13th November')
TIME: *las 11 de la mañana* ('11 am')
DURATION: *tres semanas* ('three weeks')
SET: *cada día* ('every day')

Other attributes are, for example, *value*, which makes it possible to assign a normalised value to a time expression, while *tid* gives it a unique identification number. In total there are 10 attributes in use for the TIMEX3 tag.

Additionally, TimeML builds up on the research done on events in STAG and treats also those by means of the EVENT tag. Situations that happen or occur are considered *EVENTs*. These can be punctual but also last for a period of time. Predicates which describe states or circumstances that hold as true are also seen as *EVENTs*. The LINK tag is used to represent relationships among temporal objects (events and time expressions) and distinguishes between three types of tags: TLINK, ALINK and SLINK. Important for temporal expressions are TLINKs (temporal links) as they refer

to temporal relationships and are used to annotate relationships between times, between events or between times and events. They help TimeML to anchor and order temporal objects as seen in (16).

(16) "The bus departs at 3:10pm."

```
<TLINK lid="l1" eventInstanceID="ei1" relatedToTime="t1"
signalID="s1" relType="IS_INCLUDED"/>
```

(Schilder et al., 2007:112)

The tag SIGNAL shows the relations between two entities (timex and event, event and event or timex and timex). SIGNALs generally encode temporal prepositions, temporal conjunctions, prepositions signaling modality and special characters. In (17) one can find a full annotated example in English for the sentence *John left 2 days ago*.

(17) John

```
<EVENT eid="e1" class="OCCURRENCE">
left</EVENT>
<MAKEINSTANCE eiid="ei1" eventID="e1" tense="PAST"
aspect="PERFECTIVE"/>
<TIMEX3 tid="t1" type="DATE" value="2002-07-08"
temporalFunction="true" anchorTimeID="t0">
2 days ago.
</TIMEX3>
<TLINK eventInstanceID="ei1" relatedToTime="t1"
relType="IS_INCLUDED"/>
```

These improvements on former annotation schemes have made TimeML the most complete and current state-of-the-art scheme for the annotation of time information, which has proved its usefulness already in various projects, such as in TempEval-3 (UzZaman et al., 2013).

2.2.3 Main Approaches to Temporal Parsing

The two main approaches for automatic temporal information extraction are on one side the use of hand-crafted rules and on the other side the use of machine learning techniques.

The hand-crafted rules approach uses regular expressions in order to find patterns which match to temporal expressions. Afterwards, similar regular expressions can be used to normalise the temporal expressions into a standard format.

The machine learning approach can use different techniques to fulfil the tasks. Two techniques, which are widely used in computational linguistics, are conditional random fields (CRFs) and support-vector machines (SVMs).

CRFs (Laferty et al., 2001) are undirected graphical models, a type of conditionally-trained finite state machines. SVMs (Cortes and Vapnik, 1995) are binary classifiers which are used in natural language processing for different tasks.

The state-of-the-art system HeidelbergTime (Strötgen and Gertz, 2010; Strötgen et al., 2013) is rule-based and supports various languages. It participated in TempEval-3 (UzZaman et al., 2013) for Spanish and English and outperformed the former state-of-the-art system TIPSem (Llorens, 2011) in the Spanish time expression extraction and normalisation tasks.

HeidelbergTime uses pattern files, which contain lexical markers, in order to detect time expressions within the parsing process, and additionally normalisation and rule files. The normalisation files contain information about the found patterns (as e. g. the value of a specific month), the rule files contain specific rules according to the type of time expression which is detected. HeidelbergTime makes use in the automatic annotation of the annotation scheme TimeML. So does also TIPSem (Llorens, 2011), which is the second best system in this task. TIPSem uses CRFs (for the recognition task) and SVMs (for the classification and normalisation task) for time processing and presents therefore a machine-learning approach to temporal parsing. Note that only TIPSem includes also event extraction and classification. This step is necessary for a complete temporal parsing. Thus, HeidelbergTime obtains the best results regarding the treatment of time

expressions, but TIPSem is so far the only complete system for temporal parsing in Spanish.

There exist also hybrid approaches which make use of both presented techniques. One hybrid system is KUL (Kolomiyets and Moens, 2010) which uses a maximum entropy classifier for recognising the timexes and hand-crafted rules for the normalisation part, even if it is not available for Spanish but for the English language.

The availability of systems for temporal parsing in Spanish is limited. TempEval-3 showed an evaluation of current systems and only HeidelTime and TIPSem competed for this language, while there were nine participants for English.

2.3 Discourse Parsing

The next task that is considered for a dependency model is *discourse parsing*. The model will be designed according to the needs of the task. Therefore this section presents several frameworks of importance for this specific linguistic field. It will be divided in the same way as the previous one. First, Subsection 2.3.1 explains shortly the main tasks which are implied in discourse parsing. Then, different frameworks of discourse representation are presented in Subsection 2.3.2. Afterwards, an overview of discourse parsers for English and Spanish is given in Subsection 2.3.3. The representation of discourse is very important for this project as it gives a base for the encoding of discourse information in the task-adapted discourse dependency model. The identification of this information and preparation for further use in NLP applications are fundamental for discourse parsing.

2.3.1 Discourse Parsing Tasks

Discourse Parsing consists basically of two main tasks: the division of the text into discourse units (Subsection 2.3.1.1) and the setting of the relation between those units (Subsection 2.3.1.2).

2.3.1.1 Discourse Segmentation

The first question that comes up in a discourse analysis is where to segment the discourse into smaller parts. This step is known as *discourse segmentation*. The criteria of what to consider as a segment differs in former work on discourse analysis. While there is agreement in that discourse segments are nonoverlapping, contiguous text spans (Marcu, 2000:15), segmentation is seen in different ways.

Hobbs (1985), one of the most important works on this topic, actually just segmented into sentences in his first analysis of discourse relations. This approach may have served at a first stage of research but nowadays there is a clear consensus on the use of finer-grained segments like, for example, phrasal units. This becomes clear considering the fact that specific lexical markers within a sentence, like conjunctions, can introduce discourse meaning.

Later works used different segmentation criteria to get a finer-grained structure of the text. One can find approaches which divide discourse into intentional units (Grosz and Sidner, 1986), while other use phrasal units as criterion for discourse segmentation (Lascarides and Asher, 1993; Webber et al., 1999).

2.3.1.2 Relations among Discourse Segments

Once discourse is divided into segments, the question comes up of how the different segments are connected among them and what meaning their connections imply.

The idea of relations among different parts of discourse is not new and can be traced back to Aristotle, but the basis for nowadays NLP research on discourse relations is widely attributed to Hobbs (1985) and Mann and Thompson (1986). Their work set up a number of coherence relations which have seen modifications in newer works but which can still be found as the base idea behind it.

The next subsections will explain with details the different discourse representation frameworks (Subsection 2.3.2) which have been created and the state-of-the-art discourse parsers and their approaches (Subsection 2.3.3).

2.3.2 Discourse Representations

Discourse representations can be split into two types: those that treat discourse from a textual level, and those that focus on a logic's point of view. Both have in common that they see text units not as isolated in a text, but as units which are set in relation to their context.

2.3.2.1 At Textual Level

The discourse is divided into smaller units, called discourse segments, which can correspond to full sentences but also to phrases. The aim is to connect all units according to the discourse structure and identify the meaning which is implied by these connections. The two most used frameworks for this task are introduced in this section.

- **Discourse Relations**

The coherence and structure of discourse have been an important topic of research in linguistics for a long time. In 1985, Jerry R. Hobbs wrote about this topic in the paper *On the Coherence and Structure of Discourse* (Hobbs, 1985) and established a theory which nowadays still is of high importance.

First of all, discourse is not a bunch of sentences: it has a structure. Furthermore, there are several types of coherence relations between sentences which the speaker of a language uses to construct a discourse. These different relations are motivated by the requirements of the discourse situation and are necessary to make a discourse coherent.

Hobbs states that two consecutive events in a discourse need a coherence relation other than mere succession in order to be understood and sound correct to the listener (Hobbs, 1985:9). He defines different coherence relations for discourse, which build the base for today's research in this field.

Wolf and Gibson gathered in 2006 a set of coherence relations (Wolf and Gibson, 2006) which can be seen as an improved version of Hobbs' ones (Hobbs, 1985). Therefore, in what follows the set from 2006 will be explained with some references back to the base established by Hobbs.

➤ **Temporal Sequence**

Wolf and Gibson (2006:28) describe this relation by "One discourse segment states an event that takes place before another event expressed by another discourse segment". These two discourse segments have no causal relation between their events. Hobbs describes this relation as *occasion* (Hobbs, 1985:10).

- (18) a. Primero Diana fue al gimnasio.
First, Diana went to the gym.
b. Después tuvo una reunión.
Then she had a meeting.

➤ **Cause-effect**

Wolf and Gibson subsume Hobbs' *cause* and *explanation* relations into a cause-effect relation by the use of a directed arc going from cause to effect in their graph. An example is shown in (19).

Hobbs' *cause* relation was used for discourse segments which start a cause and occur before discourse segments starting an effect. An *explanation* relation on the other hand starts with a discourse segment which expresses an effect before the next discourse segment states the cause.

- (19) a. Hubo una tormenta en Barcelona
There was a storm over Barcelona
b. y por eso no pude volar.
and therefore I could not take my flight.

➤ **Condition**

This relation is used if a discourse segment introduces an event that will only occur if the event in the other discourse segment also occurs. Hobbs (1985) did not use this relation; he used either *cause* or *explanation*. Wolf and Gibson added *conditions* as they wanted to distinguish between actual causal events and possible causal events (20).

- (20) a. Si mañana sale el sol,
If tomorrow the weather is sunny,
b. iré a la playa.
I will go to the beach.

➤ **Violated Expectation**

This relation was already used by Hobbs and corresponds to a causal relation that would normally exist between two discourse segments but is absent (21).

- (21) a. Tuvo la mejor nota de la clase,
He had the best grade of his class,
b. pero salió triste del aula.
but he came out sad of his classroom.

➤ **Similarity**

This is a relation which holds between two discourse segments with a corresponding structure (22). Hobbs called this relation *parallel*.

- (22) a. Hay un tren en el andén A.
There is a train on platform A.
b. Hay un tren en el andén B.
There is a train on platform B.

➤ **Contrast**

One can find this relation when contrasting statements are made about similar entities or when the same statement is made about contrasting entities (23).

- (23) a. Pedro le ayudó en su proyecto,
Pedro helped her with his project,
b. pero María se puso en contra de él.
but Maria turned away from him.

➤ **Elaboration**

This relation holds for two identical entities in a parallel relation as shown in (24). This relation was already presented by Hobbs (1985) and is treated in the same way by Wolf and Gibson (2006).

- (24) a. Baja la Gran Vía.
You go down Gran Via street.
b. Baja la Gran Vía y allí está al lado derecho.
You go down Gran Via street and there it is on the right side.

Additionally, Wolf and Gibson consider relations in which a second discourse segment adds new information to a first segment as *elaboration*.

Hobbs (1985) distinguished in these cases between *background* (25) and *evaluation* (26) relations. Wolf and Gibson (2006) gather all of them to *elaboration* relations and reason

this change by the difficulty in distinguishing the finer-grained categories proposed by Hobbs.

(25) a. Anna Schulz es una actriz alemana que siempre hacía malas películas.

Anna Schulz is a German actress who always did bad movies.

b. Y en 2002 ella ganó un premio por ser la mejor actriz del año.

And in 2002 she won un prize for being the best actress of the year.

(26) a. En 1930 presentaron el primer televisor.

In 1930, they presented the first television.

b. Le pareció maravilloso en aquel momento.

It seemed wonderful to him at that moment.

➤ **Example**

Discourse segments can provide examples for other discourse segments (27). Hobbs (1985) called these relations *exemplifications*.

(27) a. Existen diferentes lenguajes de programación.

There exist diffent programming languages.

b. Uno de ellos es Ruby.

One of them is Ruby.

➤ **Generalisation**

Here one of the discourse segments states a generalisation for the content of another (28). Hobbs (1985) did not use this relation but treated these as *exemplification*.

(28) a. Ayer mi ordenador detectó un virus.

Yesterday my computer detected a virus.

- b. Los ordenadores tienen muchos problemas con los ataques a través de la red.

Computers have a lot of problems with attacks through the internet.

➤ Attribution

One of the discourse segments states the source for the content of another discourse segment (29). Hobbs (1985) did not consider this relation, and also Wolf and Gibson (2006) comment that it is strictly speaking not a discourse relation but see its inclusion useful for discourse parsing purposes.

- (29) a. María dijo que
Maria said that
- b. Pablo se había comido todo el pan.
Pablo had eaten all the bread.

➤ Same-segment

This relation was not considered in 1985 by Hobbs. It is not an actual coherence relation but an epiphenomenon between two discourse segments which are actually one single segment (30).

- (30) a. La bolsa,
The stock exchange,
- b. según los economistas,
according to economists,
- c. subirá este año ligeramente.
will slightly increase this year.

Furthermore, Wolf and Gibson distinguish their established relations between directed and undirected ones and show proof that trees are not an appropriate way for the

description of discourse coherence (Wolf and Gibson, 2006:22-28). Instead they use directed graphs for their representation.

- **Rhetorical Structure Theory**

Rhetorical Structure Theory (RST) is a descriptive theory for discourse coherence. It describes the relations between text parts and provides comprehensive analyses. Furthermore, RST provides a framework for the investigation of Relational Propositions, which are inferred propositions that arise from the text structure in the interpretation of discourse (Mann and Thompson, 1986) and which are of help for the study of text coherence.

Independent to language and text type, RST uses four kinds of objects. Those are *relations*, *schemas*, *schema applications* and *structures*.

Relations hold between two non-overlapping text spans. One of the text spans is referred to as *nucleus*, and the other as *satellite*. The *schemas* use this relation to define patterns in which a particular span of text can be analysed in terms of other spans. The *schema application* conventions define how a *schema* can be instantiated. The notion of the structure of an entire text is defined in terms of composition of *scheme applications* (Mann and Thompson, 1988).

Circumstance	Antithesis and Concession
Solutionhood	Antithesis
Elaboration	Concession
Background	Conditions and Otherwise
Enablement and Motivation	Conditions
Enablement	Otherwise
Motivation	Interpretation and Evaluation
Evidence and Justify	Interpretation
Evidence	Evaluation
Justify	Restatement and Summary
Relations of Cause	Restatement
Volitional Cause	Summary
Non-Volitional Cause	Other Relations
Volitional Result	Sequence
Non-Volitional Cause	Contrast
Purpose	

Table 1: RST Relations

Table 1 shows the different relations proposed in Mann and Thompson (1988), even if they state that this is an open list which can be extended according to its usage. The indentation of items on the list means that they belong to the same subgroup. There are five relations of *cause*, for example.

According to their definition, *relations* hold between two non-overlapping text spans, which in this framework are called *nucleus* and *satellite*. A *relation* definition consists then of constraints on the *nucleus*, constraints on the *satellite*, constraints on the combination of *nucleus* and *satellite* and the effect.

Schemas define the structural constituency arrangements of text. They can be seen as abstract patterns which consist of a small number of constituent text spans, a specification of the relation between them and a specification of how certain spans (*nuclei*) are related to the whole collection.

RST recognises five kinds of schemas which are *circumstance*, *contrast*, *joint*, *motivation/enablement* and *sequence*.

There are three conventions in RST which determine the possible application of a *schema*. The schemas do not constrain the order of the *nucleus* or *satellites* in the text in which the *schema* is applied (unordered spans).

For multi-relation *schemas* all individual *relations* are optional, but at least one of the relations must hold (optional relations). And a *relation* that is part of a *schema* can be applied any number of items in the application of that schema (repeated relations).

The structural analysis of a text is a set of schema applications, so that particular constraints hold. These constraints are *completedness*, *connectedness*, *uniqueness* and *adjacency*.

The text in (31) shows an example for the Evidence relation. The corresponding diagram is shown in Figure 1 (Mann and Thompson, 1988).

(31)

1. The program as published for calendar year 1980 really works.
2. In only a few minutes, I entered all the figures from my 1980 tax return and got a result which agreed with my hand calculations to the penny.

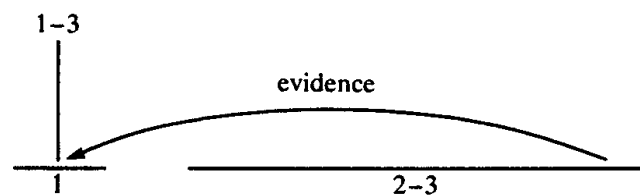


Figure 1: RST diagram for *evidence* example text (Mann and Thompson, 1988)

As can be seen in the example, Units 2-3 of sentence 2 are in an *evidence* relation with 1 and increase the reader's belief in the claim expressed in 1.

RST presents a good way to identify hierarchic structures in text, the relations which hold between its individual parts and provides a framework for the study of coherence in discourse (Mann and Thompson, 1988). Ongoing investigation on this framework can

be seen in Taboada and Mann (2006a and 2006b), Taboada and Habel (2013) and Iruskieta et al. (2015).

2.3.2.2 In Logics

After having discussed the two main frameworks for discourse representation at textual level, this subsection takes a look at discourse representations in logics.

The mapping of NLP discourse content into logic formula has been a research topic for a long time, especially since Kamp (1981) brought up Discourse Representation Theory. It is still an ongoing research topic and an automatic mapping of discourse structure and information a highly complex undertaking. The following descriptions show the most used logic frameworks for representations of discourse in logics.

- **Discourse Representation Theory**

Discourse Representation Theory (DRT) is an approach to natural language semantics and permits linguistic representations of abstract mental information. The framework was developed in 1981 by Hans Kamp and offers possibilities to represent meaning across sentence boundaries.

The human mind is capable to extract relevant features from linguistic input (text or speech) in a discourse and to use this structure in order to get to conclusions. If we want a computer to have the same language capacities as a human, it will be necessary that discourse can be analysed by the machine and that similar structures to those of the human mental representation are created.

DRT makes it possible to "look upon these processes as some sort of translations - from linguistic expressions (or their grammatical structures) to expressions of some 'language of thought' and vice versa" (Kamp and Reyle, 1993:9).

While two individual sentences like *Ana va al mercado. El vendedor da una manzana a Ana.* (‘Ana goes to the market. The shop assistant gives an apple to Ana.’) are quite easy to handle even for machines, they wouldn't be found like this within a discourse. The second occurrence of Ana would be replaced by the pronoun *le* as the hearer/reader has no problem to understand to whom *le* refers. This would result into the sentences *Ana va al mercado. El vendedor le da una manzana.* (‘Ana goes to the market. The shop assistant gives her an apple.’).

The identification of the anaphoric antecedent of the pronoun is known as *anaphora resolution*. DRT uses for its discourse representation so called Discourse Representation Structures (DRS). The advantage of DRS is that it is not only a good way for the representation of linguistic meaning but also for exploring meaning under a formal semantics approach. A DRS can be seen as a kind of non-linear (and unordered) version of standard predicate logic. This makes it also possible to convert a DRS into a formula of predicate logic. An example like *Ana compra una casa* (‘Ana buys a house’) could be presented by the DRS in Figure 2 and converted in the formula of predicate logic shown in (32).

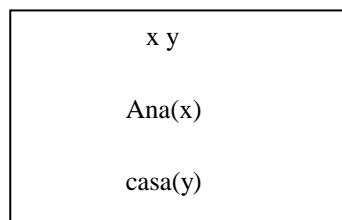


Figure 2: DRS example 1

$$(32) \quad (\text{Ana}(x) \ \& \ \text{casa}(y) \ \& \ \text{compra}(x,y))$$

Within a discourse a DRS offers the possibility to subordinate further DRSs. An example in a discourse could be, *Ana compra una casa. Le encanta.* (‘Ana buys a

house. She loves it.’). The corresponding DRS (Figure 3) and predicate logic formula (33) are shown below.

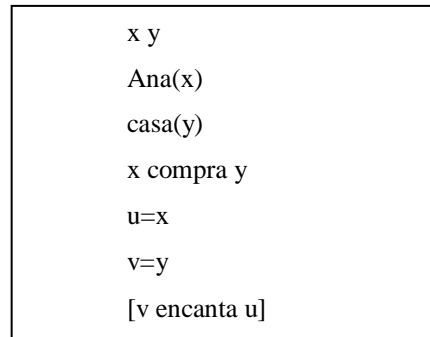


Figure 3: DRS example 2

(33) $(\text{Ana}(x) \ \& \ \text{casa}(y) \ \& \ \text{compra}(x,y) \ \& \ u=x \ \& \ v=y \ \& \ \text{encanta}(u,v))$

The conversion from the sentence into a DRS can be performed by syntactic rules, so-called DRS Construction Rules.

Let's consider the constituent syntax diagram for *Ana compra una casa* (Figure 4).

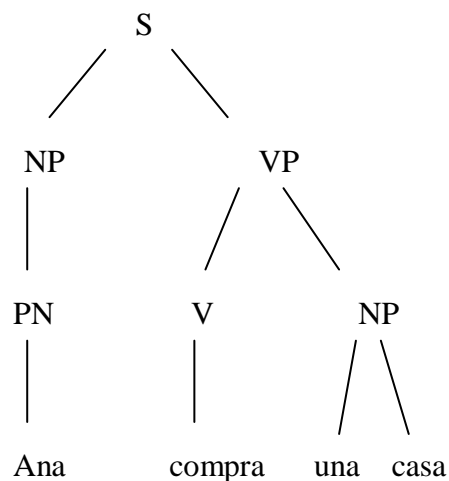


Figure 4: Constituent syntax diagram

Here the proper name Ana satisfies the predicate and so it introduces a new DRS-condition [x compra una casa]. Furthermore a DRS-condition is added which represents the discourse referent x, Ana(x). The indefinite description *una casa* (a house) is also represented by means of another DRS-condition, casa(y), and the first DRS-condition is actualised to [x compra y]. This brings us to the former mentioned DRS representation for *Ana compra una casa*.

As one can see, DRT is a powerful framework for the representation of discourse structures and an important approach in natural language processing. Note that this framework has seen its use especially in the resolution of anaphoras but not in terms of discourse relations.

- **Segmented Discourse Representation Theory**

Segmented Discourse Representation Theory (SDRT) was introduced in Asher (1993) as an extension of DRT (Kamp and Reyle, 1993) in order to account for specific properties of discourse structure.

The motivation for the extension of DRT was to cope with references to abstract objects in discourse. This means that it should be possible that the antecedent of an anaphora can be a text segment larger than a sentence. In (34), the pronoun *esto* ('this') can have as antecedent the events mentioned in *a-c* or possibly only to the last event in *c*.

(34)

a. Primero un árbol se cayó en el jardín.

First, a tree fell in the garden.

b. Después un perro murió en plena calle.

Then a dog died directly in the street.

c. Y llegando a casa, se dio cuenta que se había olvidado las llaves.

And when he got home, he noticed that he had forgotten his key.

d. Esto fue horrible para el bombero.

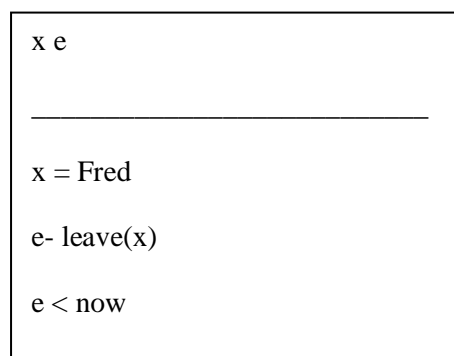
It was horrible for the firefighter.

Given that sometimes there are not constituents available within the preceding text segment, discourse relations can help to solve this problem.

"SDRT can be seen as a super-layer on DRT whose expressiveness is enhanced by the use of discourse relations." (Danlos et al., 2001)

DRSs are also used in SDRT with one difference as event references are part of the universe and can therefore be referred to. An example is given in (35).

(35)



Discourse relations can then be used for taking into account the discourse structure in the analysis (as in Lascarides and Asher, 2007).

In contrast to Rhetorical Structure Theory (RST), SDRT creates connex graphs for discourse representation, while RST uses trees for this purpose.

2.3.3 Discourse Parsers

Having discussed the different theories on discourse analysis, this subsection presents now Discourse Parsers and their approaches.

Discourse Parsing aims at the identification of relations between individual text units in order to establish a coherent discourse analysis. This has shown to be an important task

for many NLP applications such as text summarization (Louis et al., 2010; Marcu, 2000b), text generation (Prasad et al., 2005), and question answering (Verberne et al., 2007). This area sees still a lot of ongoing research and different approaches have been applied for the fulfilment of this task.

Marcu (1997) was the first well-known approach to modern discourse parsing. The algorithm used was able to detect clause boundaries within English sentences and identified rhetorical structures between them based on RST.

Both Marcu (2000a) and LeThanh et al., (2004) describe discourse parsers that make use of handcrafted rules, while the majority of later works have applied machine-learning techniques.

In 2002, Carlson et al. published RST Discourse Treebank, an RST-annotated corpus which gathers 385 documents from the *Wall Street Journal*. This presents an important resource for investigation in discourse parsing, which has seen much usage since then. Tofiloski et al. (2009) present a syntactic and lexical-based discourse segmenter and emphasise on the importance of precise discourse segmentation, as it is the first step of a successful discourse analysis.

HILDA (duVerle and Prendinger, 2009; Hernault et al., 2010) was the first fully-implemented feature-based discourse parser to work at full text level. It is trained on RST Discourse Treebank and uses RST as discourse framework. Its algorithm is based on two Support Vector Machines (SVM).

Lin et al. (2009) worked with another important discourse annotated corpus, which is Penn Discourse Treebank (Prasad et al., 2008). The corpus is a superset of RST Discourse Treebank, but without making use of RST as discourse representation. Instead they use a set of predefined discourse relations based on Webber (2004). Lin et al. (2010) presents a discourse parser based on the implied research that tries to detect also implicit discourse relations. Those are not signalled by explicit connectives, which make the identification more complicated (Pitler, 2008). The algorithm is based on machine learning and adapted for the usage of maximum entropy learning.

Wei Feng and Hirst (2012) argue that text-level discourse parsing is crucial, since two-well formed sentences do not necessarily form a coherent text. While many of former works presented discourse parsers which work and were evaluated at sentence level, Wei Feng and Hirst emphasise the importance of working at text level and comment the obviously lower performance of former parsers if they were applied to full texts. Their paper presents an improved version of HILDA with additional linguistic features included.

Joty et al. (2013) show an effective discourse parser, which makes use of Conditional Random Fields (CRF) inside the algorithm and RST as discourse representation theory. They focus on the importance of a distinction between intra-sentential (at sentence level) and multi-sentential parsing (at text level). They argue that the distinction between those two parsing levels can improve parsing performance.

Wei Fen and Hirst (2014) present a linear-up bottom-time discourse parser which beats the performance of former state-of-the-art discourse parser from Joty et al. (2013). Additionally, it works faster and implements the idea of post-edition, which can further boost the parser's accuracy. Their algorithm makes use of two linear-chain CRFs with additional constraints in the Viterbi decoding of the first one.

As seen, discourse parsers see nowadays a high use of machine learning techniques, which make use of annotated discourse corpora. The number of those corpora is limited and therefore the available resources see usage in different parsing projects.

Discourse parsing for Spanish is still in its beginnings. Da Cunha et al. (2011) presented with RST Spanish Treebank a discourse-annotated corpus. It gathers 52,746 words and represents an important resource for the investigation of discourse parsing in Spanish. Afterwards, the next steps were the creation of a system for Spanish Discourse Segmentation (da Cunha, 2012a) and a system for assigning nuclearity and rhetorical relations based on RST Spanish Treebank (da Cunha et al., 2012b). Da Cunha (2013) comments on the difficulty of resolving ambiguity of discourse markers and the plan to implement a full discourse parser for Spanish in the future.

It is worthy to comment that there is currently work on discourse parsing in both Basque and Portuguese. Iruskieta and Zafirain (2015) present a dependency-based approach to discourse segmentation for Basque and Iruskieta et al. (2015) a related work on the detection of central units. Pardo and Nunes (2008) explain the creation of a discourse parser for Brazilian Portuguese.

3. Methodology

The idea of this dissertation is to create NLP task-based dependency models by automatic means. This shall be done by the adaptation of a base dependency corpus according to the needs of each model. It is planned to create task-based models for two different NLP tasks. The following sections describe the ideas, resources and tools on which the following investigation is based. First the used corpus, which serves as starting point for this investigation, is presented (Section 3.1), afterwards ideas for the creation of different dependency models are introduced (Section 3.2) and, finally, the evaluation method is explained (Section 3.3). Note that the examples throughout this doctoral thesis are mainly taken from AnCora and from the website of the Spanish newspaper El País. In order to provide examples with simple structures, which clearly explain the discussed linguistic phenomenon, there are also some examples which do not come from the main sources but are added after proofreading by native speakers.

3.1. Corpus

AnCora has been chosen as starting point, since it offers a multilevel annotated corpus for Spanish at an important size and includes a dependency layer. This layer turns out to have several problems for the use in this project but the constituent layer makes it possible to create a new surface-syntax dependency annotation which is an adequate base for the present investigation.

The following subsections explain the AnCora corpus creation (Subsection 3.1.1), take a closer look at the included dependency layer (Subsection 3.1.2) and show why a new dependency layer for the corpus is necessary as base data for this project (Subsection 3.1.3).

3.1.1 AnCora Corpus Description

In 2008, Taulé, Martí and Recasens presented the AnCora corpus, a linguistic work which had already been started years before. In 2004, first steps were done under the name 3LB (Civit and Martí, 2004), a corpus which was used to build up to AnCora together with CESS-ECE (Martí and Taulé, 2007). Both corpora included mainly newspaper and newswire articles.

AnCora offers nowadays the largest multilayer annotated corpus for Spanish and Catalan and is freely available². The corpora have already seen use in several international evaluation competitions such as CoNLL-2006, CoNLL-2007 and SemEval-2013 for different syntactic and semantic NLP tasks.

The Spanish version AnCora-ES contains 517,269 words, while the Catalan version contains slightly less with 488,389.

Corpus	Tokens	Sentences
AnCora-ES	517,269	17,376
AnCora-CA	488,389	16,591

Table 2: AnCora corpus size overview

Both corpora were annotated for several layers of linguistic information. This includes one layer at morphological level, different levels of syntactic descriptions (including constituent and dependency layers) and also semantic information was added. Depending on the task, the annotation was fulfilled manually, semiautomatically or completely automatically.

The corpora have an automatic morphological tag annotation and disambiguation (Civit and Martí, 2004) which was manually revised. A manual annotation was also added for deep syntactic information, namely constituents and functions, and strong and weak named entities. Additionally, WordNet nominal synsets were annotated in the same way.

² under <http://clic.ub.edu/ancora>

A semantic annotation of verbal predicates was performed in a semiautomatic way. Thematic roles were automatically associated with the syntactic functions. This was done based on the verbal lexicons AnCora-Verb-Es and AnCora-Verb-Ca. These lexicons make explicit the mapping between syntax and semantics (Aparicio et al. 2008). Manually written rules mapped automatically the information of the lexicons onto the syntactic structure. This made it possible to add thematic roles and semantic classes. The process was later on revised manually.

3.1.1.1 Morphological Information

AnCora distinguishes the part of speech (POS) and minor morphological categories such as gender, number, case, person, time, and mode. This information is coded into a character sequence following the EAGLES proposal (EAGLES, 1996), as exemplified in Figure 5. The first character corresponds to the main category (e.g. noun) and the second to the subcategory (e.g. common noun). The next five characters and digits were used according to the category and are optional, as some categories (e.g. punctuation) only make use of the first two characters.

Word	Lemma	POS
Si	si	CS
trabajo	trabajar	VMIP1S0
bajo	bajo	SPS00
presión	presión	NCFS000
bajo	bajar	VMIP1S0
el	el	DA0MS0
interés	interés	NCMS000
.	.	Fp

Figure 5: Morphological information example

AnCora uses a total of 280 different labels here, but the size is reduced to only 47, if only the first two characters are considered.

3.1.1.2 Syntactic Information

The AnCora corpora started with constituents and functions, and included elliptical subjects. The dependency annotation was later added by a conversion process (Civit et al. 2006). As the syntactic information layers are the important ones for this project, Subsection 3.1.2 will give a detailed look on AnCora's dependency annotation.

3.1.1.3 Semantic Level

The semantic annotation consists of the argument structure of verbal predicates and their semantic class. The information was coded into the following labels:

Arg0, Arg1, Arg2, Arg3, Arg4, ArgM, ArgA and ArgL.

The first five are numbered from less to more oblique regarding the verb, ArgM refers to adjuncts, ArgA to external agents and ArgL to complements of light verbs, which are often lexicalised as *un beso* ('a kiss') in *dar un beso* ('give a kiss'). Thematic roles are coded into 20 different labels, such as *agent*, *cause*, *experience*, etc. Weak and strong named entities were also annotated for both AnCora corpora (Borrega et al. 2007). *Weak named entities* refers here to different kind of noun phrases, while *strong named entities* refer to proper names such as personal names, book titles or country names. The information was encoded in the POS tag. Finally, a WordNet sense was added to each noun, which was done manually and by help of Spanish and Catalan EuroWordNets-1.6.

3.1.2 Dependency Annotation

As mentioned previously, the AnCora dependency annotation was added after the initial creation of constituents structure by means of a conversion from those (Civit et al. 2006). The conversion was done automatically but with manually written head and function tables.

The decision to convert from constituents to dependencies was taken since the constituent annotation is richer in detailed information. Therefore, it is a straightforward conversion into dependencies, which are more compact, but would be non trivial the other way around as information would have to be added within the conversion process.

Both constituent and dependency data were made available later on for research purposes. The resulting dependency annotation consisted then of the data as exemplified in Figure 6. This includes information about the position in the sentence, word form, lemma, part-of-speech, head, syntactic function, first part-of-speech letter and further linguistic information of each token in the corpus.

1	Las	el	DA0FP0	2	spec	d	gen=f num=p postype=article		
2	reservas	reserva	NCFP000	9	suj	n	gen=f num=p postype=common		
3	de	de	SPS00	2	cn	s	postype=preposition		
4	oro	oro	NCMS000	3	sn	n	gen=m num=s postype=common		
5	y	y	CC	4	conj	c	postype=coordinating		
6	divisas	divisa	NCFP000	4	grup.nom	n	gen=f num=p postype=common		
7	de	de	SPS00	2	cn	s	postype=preposition		
8	Rusia	Rusia	NP00000	7	sn	n	postype=proper ne=location		
9	subieron	subir	VMIS3P0	25	cd	v	num=p postype=main person=3 mood=indicative tense=past		
10	800	800	Z	11	spec	z	ne=number		
11	millones	millón	NCMP000	9	cd	n	gen=m num=p postype=common		
12	de	de	SPS00	11	sp	s	postype=preposition		
13	dólares	dólar	Zm	12	grup.nom	z	postype=currency ne=number		
14	y	y	CC	9	conj	c	postype=coordinating		
15	el	el	DA0MS0	16	spec	d	gen=m num=s postype=article		
16	26_de_mayo		[?:26/5/?:?:?:?]	W	18	cc	w	ne=date	
17	_	_	__elliptic__	18	suj	sn	_		
18	equivalfan	equivaler	VMII3P0	9	S	v	num=p postype=main person=3 mood=indicative tense=imperfect		

Figure 6: AnCora dependency annotation example

Regarding linguistic decisions, Civit et al. (2006) comment that the manually written rules for the head selection were linguistically based, and they give information about the treatment of coordinations. While in most Spanish structures it is quite clear which word should be the head of a specific structure, coordinations are more complicated. A coordination structure normally contains a conjunction and two or more coordinated elements. Different solutions can then be adopted for the dependency annotation. Prague Dependency Treebank (Hajic, 1999) takes the conjunction as head, AnCora dependencies (Civit et al., 2006) prefer to have the first coordinated element as head of the coordination structure.

In an example like *Comieron y hablaron* ('They ate and talked'), both the conjunction *y* ('and') and the second verb *hablaron* ('they talked') would be dependent of *Comieron* ('they ate') according to AnCora dependencies, while the conjunction would be the head of both verbs in the other approach.

Civit et al. (2006) do not give detailed results regarding the conversion into dependencies but state that this task can be achieved with a high accuracy and that it was a good way to check the quality of the previous constituent annotation.

The tagset, which is used in an automatic conversion, is important both for the expressiveness of the data and for the quality of the annotation. A high number of tags may give more specific linguistic information about the data as a reduced number of tags, but on the other hand the quality of an automatic conversion can suffer if too many tags are used. AnCora made use of 51 tags, which is a reasonable quantity for an automatic conversion.

Table 3 shows the used syntactic function tags in AnCora dependencies and their frequencies in the corpus.

<u>Tag</u>	<u>Frequency</u>
interjeccio	59
grup.adv	52
ci	3042
voc	13
sentence	17393
cc	35188
cpred	1976
participi	4
ao	2484
cd	27875
morfema.verbal	47
pass	1789
grup.verb	3
et	1341
conj	10122
n	137
inc	524
i	10
neg	273
morfema.pronominal	3032
s.a	25174
subj	41584
impers	456
relatiu	333
z	897
cag	1397

sadv	2652
mod	5073
a	146
c	427
prep	9
d	2420
f	65289
infinitiu	499
grup.nom	3986
sp	43502
gerundi	9
coord	15104
spec	78265
p	38
s	1284
r	279
atr	6430
sn	83824
w	25
v	9281
espec	1
sa	136
grup.a	742
creg	5294
S	28594

Table 3: AnCora dependency function tagset

The first idea for this dissertation was to take the AnCora dependencies as base data for the planned automatic creation of task-oriented dependency models but some issues with the data made it necessary to reconsider this idea. These issues are explained in the next subsection.

3.1.3 AnCora Dependencies Issues

Several problems were detected while considering AnCora dependencies as base data for the upcoming model creation.

First of all, AnCora works with multiword tokens, an artificial construct that is not reflected in the surface of the text. Thus, instead of *Universitat de Barcelona* ('University of Barcelona'), *Universitat_de_Barcelona* is found in AnCora. This treatment is not limited to compound nouns, it can be found through all kind of linguistic structures. Chapter 4 will give further details. A semantic approach within the dependency annotation is the reason behind the use of multiwords. Multiwords are supposed to make information extraction easier, while reducing the sentence complexity in parsing purposes. Nevertheless, several problems appear with this approach if used as base version.

It may not always be useful to have certain structures written as multiwords, and the treatment has to be consistent or it can introduce errors in the annotation that are difficult to handle. A conjunction like *mientras que* ('while'), for example, should not be found within the corpus both as two tokens and written together as multiword. Additionally, most NLP parsers do not work with multiwords and this would make an adaptation of the data necessary.

The general semantic approach in AnCora dependencies, which includes head selection, is also a point that is not really advantageous for the planned base version. Many NLP applications do rely primarily on semantic information (e.g. information extraction) but others do not (e.g. grammar checker) and it will always be easier to adapt a syntax-based annotation to a semantic approach than the other way around. Linguistic information like agreement between predicate and subject is found in the auxiliary verb in composed structures like *han ido* ('they have gone') and not in the verb which carries the semantic information.

This brings us to the next arguable point, which is the treatment of coordinations. As a syntax-oriented base version is preferred for the upcoming dependency customisation, it is difficult to work with the coordination treatment chosen for AnCora dependencies. If

the first coordinated structure is taken as the head of a coordination structure, then in an example as *el chico y la chica cantan* ('the boy and the girl sing') *chico* ('boy') would be the head of the coordination structure, and thus be directly connected to the verb *cantan* ('sing'). Agreement between subject and predicate is not given in this case. The other proposed (but not implemented) option by Civit et al. (2006) to treat the conjunction as head would definitely be the better choice for a syntax-based treatment. In the previous example both coordinated nouns are singular in terms of number on their own, but only joined together they show agreement with the verb.

Another point which should be commented here is that the inclusion of elliptic subjects in AnCora dependencies was neither seen as a good idea for a purely syntax-based base version. An example like (36) would cause the introduction of such an elliptic subject in AnCora.

- (36) Come pan.
She eats bread.

This inclusion is reflected in the annotation by the introduction of an empty token at subject position, which is represented by an underscore as word form. Both multiwords and elliptic subjects are artificial constructs that are introduced into the text. The addition of elliptic subjects is also not really important for parsing purposes since Spanish verbs contain this information implicitly.

The presence of these issues in the AnCora dependencies made it necessary to create a new fully syntax-based dependency annotation, which will be presented in Chapter 4.

3.2 Dependency Customisation

In the previous section a corpus has been presented which can be used as starting point for the planned investigation. Now, it is time to have a look at possible ways to conduct an automatic dependency customisation for specific NLP tasks. Temporal and discourse parsing have been chosen as such tasks, but, before this dissertation can start to focus on

these tasks, there are still some basic ideas to discuss. Thus, this section explains the main ideas behind the task-based customisation of dependency structures (Subsection 3.2.1) and describes a possible way to evaluate the *optimisation*³ between non-adapted and a task-adapted dependency structures (Subsection 3.2.2).

3.2.1 Main Ideas

Linguistic corpora are a limited resource. Nowadays many projects work with dependency annotations in order to obtain syntactic information (Levy and Goldberg, 2014; Ma et al., 2015). The applied linguistic criteria in an annotation can vary and focus on either syntactic or semantic language features. This does not mean that there are only two possible annotation types. While those are the two basic directions, the annotation criteria can lie in between those endings according to the desired individual choices. The implementation of the chosen linguistic criteria can be done by means of head selection within the dependency structures. Chapter 1 of this dissertation showed an example with different words of interest according to the further usage. Taking another look at it in (37), one can see that by different choices in head selection relevant nodes for the further usage can be better positioned. In (37c), the version oriented to discourse parsing, it is important to decide how to connect the conjunction within the dependency structure. On the one hand, it contains important information itself, as it puts into relation main and subordinate clauses; on the other hand, a discourse parser needs to extract the event information of both clauses and a short distance between the verbs is an advantage.

- (37) a. Valentina *viaja* en tres semanas porque ha ganado el premio.
Valentina travels in three weeks because she has won the prize.
- a. Valentina **viaja en tres semanas** porque ha ganado el premio.
- b. Valentina **viaja** en tres semanas **porque ha ganado** el premio.

³ The term *optimisation* is used throughout this chapter referring to a measurable improvement of a network's performance, as in Newman (2010:541-551).

Chapter 2 has shown state-of-the-art regarding the two chosen NLP tasks for this dissertation: temporal and discourse parsing. Both NLP tasks make use of specific annotation schemes and theories to mark explicitly relevant information for the tasks. For example, a temporal parser needs to know about time expressions (37b), and a discourse parser about conjunctions, which introduce discourse-related meaning (37c). This has to be taken into account in the creation of specific dependency structures in order to position the relevant nodes in better positions.

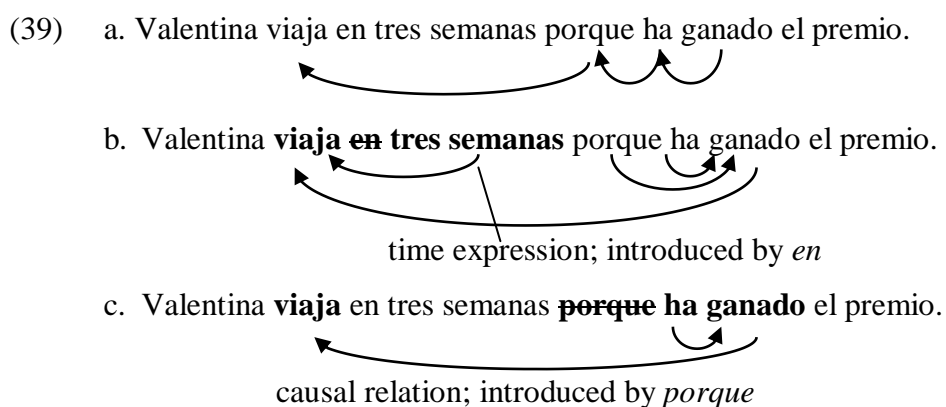
Additionally, dependency annotations include another layer of standard information: syntactic function tags. The tag in a dependent-head relation of two nodes in a dependency structure states the relation the dependent has regarding its head. For example, AnCora marks the relation between a subject and its corresponding verb as *suj*. The tagset encodes therefore relevant information for a dependency structure. A task-based dependency model can use this tagset to enrich a dependency annotation with specific information. In (37b) *semanas* could be marked for the parser as a time expression and in (37c) *porque* could hold information about its discourse-related meaning.

Furthermore, a task-adapted head selection can take into account the specific preferences of an NLP task and offer short paths through the dependency structures for the extraction of relevant information. Example (38) shows different dependency structures based on the same sentence but analysed under a different focus. (38a) shows a surface-syntactic analysis of the sentence, while (38b) presents an adapted version for a temporal parser and (38c) a version adapted for a discourse parser.

- (38) a. Valentina *viaja en tres semanas porque ha ganado el premio.*
- b. Valentina **viaja en tres semanas** porque ha ganado el premio.
- c. Valentina **viaja** en tres semanas **porque ha ganado** el premio.
-

The different versions of dependency structures offer already shorter paths to the relevant information according to the chosen focus. This has been done so far by means of head selection. The enhancement of information by means of the tagset gives another possibility to tailor a dependency structure according to the needs of its further usage. In (38a), for example, *ha* ('has') is the head of *ganado* ('won') and the latter would have a syntactic function tag like "verbal object" to indicate the relation to its head. In (38c) the head-dependent relation is inverted and this would be reflected by the syntactic function tag. This would then indicate that *ha* is an auxiliary verb, which connects to *ganado*. A parser can access the tag easily and extract the given information.

Another possibility to model dependencies and to further shorten distances to the relevant information is the use of collapsed dependencies (de Marneffe and Manning, 2008). Example (39) shows how their application can affect the former analyses seen in (38). The tagset can hold the information given by disconnected nodes. In (39b), for example, the tag of *semanas* ('weeks') can indicate that it is a time expression, which is introduced by the preposition *en* ('in') and in (39c) the tag of *ganado* ('won') can hold the information that it is in a causal relation to *viaja* ('travels'), which was introduced by the disconnected conjunction *porque* ('because'). In this way, relevant information gets a compact representation and a parser has only to walk short distances to find it.



The former examples have shown different ways to present a dependency analysis for one and the same sentence, and the different means that can be used to adapt a dependency structure to its further usage.

Note that there is a whole range of NLP tools which can benefit from dependency annotations, and that the linguistic features, which are important for each task, are very different. A grammar checker will take advantage of a purely syntactic annotation, an automatic summariser of a text has to extract semantic information, and a temporal parser, as seen before, will additionally need to know about time expressions.

Now, considering the variety of NLP applications, it is difficult to imagine that there exists something like one single “perfect” dependency annotation. It makes sense to add task-relevant information to a dependency annotation, but the criterion for what is considered as relevant is very different according to the task. There is no reason why a time expression should be marked in a special way for a discourse parser, and the addition of unnecessary information cannot be a desired choice for a dependency annotation. Furthermore, it is impossible to improve the positioning of all nodes in a dependency structure. If one node improves its position, another node declines. So, it will not be possible to adapt the head selection criteria to all possible further usages of a linguistic corpus.

The present dissertation suggests therefore a way to automatically adapt the input data to the desired further use by applying a specific treatment on a basic surface-syntax dependency annotation, which includes the necessary linguistic criteria and information. NLP tools will then later on directly use an adapted input with rich information according to their tasks. The necessary program to perform the adaptation to the chosen NLP tasks (temporal and discourse parsing) will be written as part of this thesis and presents an innovative approach to dependency parsing and automatic dependency modelling.

3.2.2 Evaluation Approach

The evaluation of the specific models has to be done at several levels. First of all, it has to be seen if the adaptation itself proceeds successfully. This is important as the adaptation is a fully automatic process and it has to be ensured that it obtains the desired results. Afterwards, it has to be seen if and how the optimisation from the surface-

syntax dependency model to the task-specific models can be expressed in numbers. There are several options to measure this optimisation. One or several NLP tools could be used with the surface-syntax base corpus and the optimised annotation. The performance of the respective tool could then be evaluated in order to see if it improves. This would be a practical approach. Another possibility is to undertake an evaluation at a more theoretical level. Networks have an important place in computation research and the creation of language networks by means of the different corpora (Ferrer i Cancho, 2004) could be an interesting undertaking. This can make it possible to investigate language from a mesoscopic point of view, where words act as basic units which interact according to grammar rules (Choudhury and Mukherjee, 2009:1). While there will be most probably opportunities to see the created corpora of this doctoral thesis in practical use, once published, it is unlikely that there will be a better occasion to conduct the network-based evaluation approach. Therefore, it was decided to choose the more theoretical evaluation approach as it promises to have more benefits for the research community.

3.3 Network-based Evaluation

The idea is to establish a language network based on a purely-syntax oriented base corpus and compare it to networks created out of the task-specific model corpora. Nowadays language networks are still an ongoing research topic (Solé et al., 2010; Čech et al., 2016) and therefore it does not exist a clear indication of how such an evaluation has to be conducted. So, one goal for this thesis will also be to prove several measures of network importance for their adequateness to evaluate language networks.

General research on networks has already established several measures of importance and tools (such as *Igraph* or *NetworkX*), which can calculate those measures automatically. Thus, the challenge will not be their calculation but to interpret them according to the dissertation's context. They should also show proofs for the optimisation process performed between the different corpus versions.

3.3.1 Introduction

Networks are an adequate way to represent and investigate data in many fields of research. Social networks can be analysed with the help of such a representation, and also physical or biological networks like the Internet or the human brain. The present research will focus on another type of networks – to be more concrete on the syntactic composition of human language.

A network representation is normally done by means of its basic components, which are *nodes* (also called *vertices*) and *edges* (also called *lines*) in order to connect them. These *edges* can be *directed* or *undirected*. An *undirected edge* can be crossed in both directions while a *directed edge* indicates the direction from which it can be crossed and offers in this way a one-side connection.

If one thinks in terms of social networks, a directed graph would correspond to a situation in which one user is connected to another user but not the other way around. An example would be the short message based Twitter with user A following user B without having user B following user A. An undirected graph would imply that both users are connected with each other. This could be, for example, a friendship between two people on the social network Facebook. In terms of websites one can think of a link from page A to page B as an example for a directed graph and if page B also links to page A for an undirected graph.

The optimisation of all kinds of networks is very important as those optimisations transfer later on to better results of the implied components. An internet connection will be faster if the less cost expensive route through the network is chosen. Users of social networks like Facebook and Twitter are fighting for power in terms of connections to other users as it facilitates publicity or even electoral campaigns. These two examples give also a first idea of two basic evaluation concepts of networks, the number of connections a node has within it and the distance one node has to travel in order to reach another node.

Nevertheless, these are not the only interesting results one can observe in a network analysis. Scientists of a variety of fields have worked over many years to create tools

which gather the most important calculations that can be made and help to interpret a network structure.

3.3.2 Research on Language Networks

Over the last two decades more and more interest has been found by the research community in the investigation of language features by means of complex networks. These can help to do research on language from a mesoscopic point of view, where words act as basic units which interact according to grammar rules (Choudhury and Mukherjee, 2009:1). The results can be useful at theoretical level and help to investigate structural properties of language, which can then support linguistic theories. On the other hand an applied use in natural language processing can be seen as in the induction of syntactic categories (Harris, 1968), word sense disambiguation (Galley and McKeown, 2003), or information retrieval (Page et al., 1998). The content represented in language networks can vary according to its design and the specific research purpose it is created for. If one wants to investigate the accessible word forms in a human brain, a mental lexicon can be created, which is a network containing all those forms. Miller and Gildea (1987) prove, for example, that an average high school student already has a receptive vocabulary of over 100,000 words. Considering that this network is considered by a person in milliseconds, it shows in a good way what a difficult task the human brain is performing and gives an idea why the understanding and controlling of this kind of network will have positive influence on areas such as (foreign) language acquisition. Examples of works based on mental lexicons can be seen in Vitevitch (2004), Tamariz (2005) and Kapatsinski (2006).

This should not lead to the conclusion that language networks are mainly focused on mental lexicons. There are many ways to prepare them according to different research interests. Wordnet (Miller, 1995) is a famous example for a network based on semantic similarities. Its nodes represent concepts which are connected by semantic relationships, and it finds wide use nowadays in linguistic projects, such as in Patwardhan and Pedersen (2006). Other research areas for language networks can be found in word co-

occurrence, as in Choudhury et al. (2010), and also in phonological networks, as in Mukherjee et al. (2007).

Nevertheless, the focus of this section is on syntactic dependency networks, as this is the topic of the present project. Following dependency grammar, two words are connected by a directed relationship that corresponds to head-dependent pairs. This relation can easily be reproduced in a network as pairs of nodes are connected between each other by means of directed edges (Choudhury and Mukherjee, 2009:7). These edges can then be used to express specific information (such as syntactic or semantic information of this relation). So a language network based on syntactic dependencies can be created by means of a dependency-annotated corpus. This is done by connections in the network between all words that have been syntactically linked in a sentence within the corpus at least one time (Ferrer i Cancho, 2005:65). From this moment on, a dependency-annotated corpus is no longer investigated at sentence level but as a whole.

This different point of view makes it possible to observe characteristics of a certain language or of a specific annotation, which at sentence level would be unobservable. Ferrer i Cancho et al. (2004) showed the presence of the small-world phenomenon for a Romanian dependency network. It was observed that any individual word of the network could be reached by an average of 3.4 edges. Another property of syntactic dependency networks is a heterogeneous degree distribution. This means that many words in the network have only few connections, but the proportion of words with many links is significant (Ferrer i Cancho, 2005:66).

It was also shown in Ferrer i Cancho et al. (2004) that the relationship between word frequency and word degree is approximately linear. Thus, the distribution of word frequencies could directly affect the distribution of word degrees. Zipf's law for word frequencies (Zipf 1935 and 1949) shows also approximately the same exponent as the value for word degree distribution. Ferrer i Cancho (2005) comments that the latter distribution could be a side effect of the associations with meanings in a communication system.

Generally speaking, most research has been done with English as working language (Ferrer i Cancho and Solé, 2001; Motter et al., 2002; Sigman and Cecchi, 2002;

Mitchell et al., 2008). Ferrer i Cancho et al. worked (2004) with dependency data from three different languages (Romanian, Czech and German). Nevertheless, no work has been done on Spanish dependency corpora so far.

The observations made in research also give direct insights into the language handling of the human brain. The small-world phenomenon gives an explanation why the human being is able to create and understand utterances in such a fast way. All words inside the person's language resources are connected very well by only a few intermediates. The heterogeneous degree distribution could explain what is happening in cases of agrammatism (Ferrer i Cancho, 2005:67). This form of expressive aphasia (Caplan, 1987) shows the omission of function words in human speech which could be explained by the fact that networks with a power degree distribution are very sensitive to the disconnection of the nodes with most connections (Jeong, 2002).

Ferrer i Cancho (2005) underlines the resemblance of the human brain's structure and the structure of a global syntactic dependency network. He also comments that syntactic dependency based formalisms seem to be a better approach for modelling human language as the use of classic phrase structure models (Chomsky, 1957) and its later developments (Chomsky, 1995; Uriagereka, 1998).

Čech et al. (2016) underline the importance of syntactic networks investigation for language research, but also describe some shortcomings of the research done so far. They argue that the global properties of language complex networks are due to word frequency rather than syntax (Čech et al., 2016:184) and that previous research (such as Solé, 2005; Solé et al., 2010; Corominas-Murtra et al., 2010) gave too much credit to syntax for observations as small-world-likeness in this type of networks. Furthermore, Čech et al. (2016:179-181) show that the analysis of syntactic networks is a task which is difficult to compare to other complex networks (such as for World Wide Web connections) and which highly depends on the syntactic annotation criteria. As future challenges, they highlight the need for more in-depth linguistic interpretations in syntactic networks, a detailed analysis of the impact of syntax on network characteristics, comparable results of parsing formalisms (or a universal formalism), and the relationship between syntactic and cognitive networks (Čech et al., 2016:185).

The next subsection will present basic concepts of network analysis and their significance for the final evaluation of this dissertation.

3.3.3 Network Concepts

The mathematical representation of networks can be conducted in different ways; normally the nodes are labelled with unique identifiers such as names or numbers and an edge list contains all connections between those.

This will be enough information to create a network if all of its paths have the same importance. However, there are also cases where their importance has to be established explicitly by means of weights. An edge list without weights could look like Table 4 and one with weights like Table 5.

1	2
2	3
3	4

Table 4: Edge list without weights

1	2	65
2	3	34
3	4	7

Table 5: Edge list with weights

The first column indicates the first node, the second column the second node of the edge and, in case there is a third column, it introduces the weight of this edge.

When working with a language network, it makes sense to take word names or lemmas instead of numbers as identifiers of the vertices. An example would then look like Table 6 and Table 7.

ir	haber
haber	a
a	volver

Table 6: Lemma edge list

ir	haber	65
haber	a	34
a	volver	7

Table 7: Lemma edge list with weights

In an Internet network these weights could refer to the amount of data sent through an edge and, in a language dependency network, to the frequency of dependent-head pairs. Furthermore, it is important to distinguish between directed and undirected networks. While an edge in an undirected network is traversable from both sides of its vertices, an edge in a directed network indicates in which direction the information can flow.

If the example in Table 4 represents an undirected network, “1 2” is the same as “2 1”, while in a directed network “1 2” means that there is only a way from 1 to 2 and not necessarily the other way around. Note that this difference is important for the work with dependencies. A dependent-head pair corresponds normally to a directed edge as it indicates explicitly which of the two is the head and which is the dependent.

A path is a route across the network from one node to another and its length can be defined as the number of traversed edges within it. A special case, and many times the most interesting, is the *geodesic path*, which refers to the shortest path in the network between two nodes. In the calculation of the shortest path it is important to consider if the network is directed or undirected as this affects highly the routes that can be taken. The *average path length* in a network is one of the basic general network measures, and the mean distance for a specific node to travel to other nodes of the networks is also a measure of importance at node level (see Subsections 3.3.4.1 and 3.3.4.2).

Two other concepts in a network that should be mentioned here are *hubs* and *authorities*. The term *hub* refers to nodes regarding its number of outgoing connections, while *authority* refers to a node regarding its number of incoming connections. Both can be regarded as measures of importance for a node within the network and are individually measured as *in-* and *out-degree* or combined as *total-degree* (see Subsection 3.3.4.2).

3.3.4. Network Measures

During many years researchers of different fields have gathered a set of important network measures which are helpful for the interpretation of the created networks. If one takes a look at these measures, a first division can be made between network-level and node-specific measures. Former measures give us information about general characteristics of the network itself, while the latter describe single nodes and help to interpret their importance in the network.

3.3.4.1 General Network Measures

Several measures give information about the general characteristics of a network. A network can be overall constructed in a manner that information gets from individual nodes to other nodes of the network in the shortest possible way. This can be imagined like a road net within a city, where the government tries to optimise routes through the city so that employees get to their work in the shortest time possible. The same should happen in an optimised network. The measure which takes this into account is called *average path length*.

Another measure which gives information about the network is the *clustering coefficient*, also called *transitivity* (Newman, 2010:198). It shows the degree in which nodes in the network tend to cluster together. Specially in social networks, this value shows a high importance since a person is more likely to be the friend of a friend rather than to be a friend of a random person.

The third general measure which is considered is *assortativity*. The term refers here to the tendency of nodes to connect to nodes with a similar degree to their own (Newman, 2010:266). The result is measured by a value between -1 and +1, where +1 would completely confirm this tendency and -1 represent a complete disagreement with it. Positive values are referred to as *assortative mixing* and negative values as *disassortative mixing*. Social networks have normally positive values, while neural networks tend to show negative values (Newman, 2010:237). Ferrer i Cancho et al. (2004:3) have presented results showing disassortative mixing also for language networks based on syntactic dependencies.

3.3.4.2 Node-specific Measures

A range of measures do not tell us something about the network's general characteristics but give us information about specific nodes in the network. This will be normally related to their importance within the network. But *importance* can refer to many different points of view.

First of all, a node is important if there are many other nodes in the network pointing to it. This situation corresponds to a web page which receives many links from other web pages or a word which is head of many other words in a dependency corpus. But outgoing connections can also be considered as a sign of importance. The former value is commonly known as *in-degree*, and the latter as *out-degree*. If one wants to refer to both values in a single measure, this can also be done by taking the sum and making it the *total-degree*. A finer use of the concepts of incoming and outgoing connections of a node is also made by *Katz centrality* (Newman, 2010:172) and by the famous Google algorithm *PageRank* (Page et al., 1998).

Nevertheless, the number of connections a node has within a network is not the only measure of importance that can be calculated. If one imagines information travelling in the form of packages through a network, then it is clear that a node which lies on the way of the shortest path of two nodes is important for the arrival of the package. If information travels from node A to C through B, then a disruption of B would cause

serious problems. Node B is therefore of importance for an optimal functioning of the network. This type of importance can be measured with the help of *betweenness* (Newman, 2010:185).

The average path distance was already mentioned as a network-level measure, but the mean distance of a node to other nodes also provides node-specific information. A node will be well situated within a network if the distance to other nodes is short. This can be measured with the help of *closeness centrality* (Newman, 2010:181).

- **In-Degree**

In a directed network, as it is the case in the final evaluation, the *in-degree* refers to the sum of all incoming connections from other nodes for a certain node in the network. A node with a high in-degree is also referred to as *authority*. In the final evaluation of this project this value refers to the number of nodes which have a specific node as dependency head.

In (40) *casa* ('house') would have an in-degree of 3. The determiner *la* ('the') and the adjective *verde* ('green') are its dependents in the first sentence, which takes the in-degree to 2. The second sentence adds an additional point to the in-degree as again the determiner has *casa* as its head.

- (40) a. El chico va a la casa verde.
The boy goes to the green house.
- b. La casa es grande.
The house is big.

Figure 7 shows a network representation based on the example seen in (40). Note that in a syntactic dependency network the linguistic criteria for the head selection of the dependency relations have a direct effect on the composition of the network.

Furthermore, note that the nodes of the network in Figure 7 refer to the lemmas of the example.

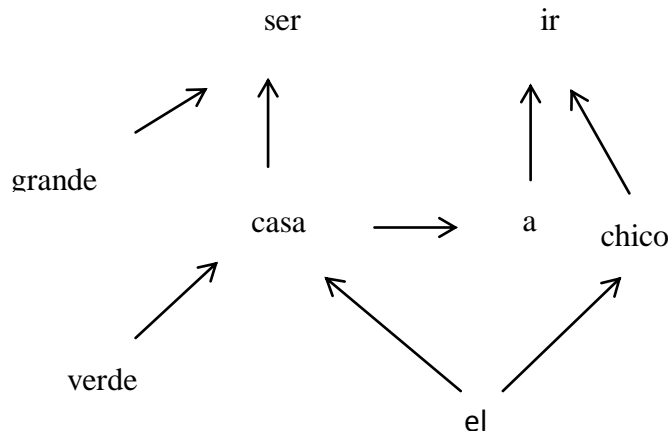


Figure 7: Example for a syntactic dependency network

- **Out-Degree**

This value refers to the number of outgoing connections of a node in the network. In the final evaluation of this project, this refers to the number of different heads a node has. A node with a high number of outgoing connections is considered a *hub*. The example shown for *in-degree* in (40) can also be used to exemplify the *out-degree*. The node *casa* has one head in the first sentence and one head in the second sentence and therefore an *out-degree* of 2 in the example.

- **Total-Degree**

The *total-degree* is the combined measure of *in-* and *out-degree*, so the sum of both individual values. In terms of a dependency network, the measure is referring to the sum of all dependents and all heads of a specific node. In example (40), the *total-degree* for *casa* would be 5 as it is the sum of the *in-degree*, which is 3, and the *out-degree*, which is 2.

- **Eigenvector Centrality**

This measure can be seen as an extension to *degree centrality*. While *degree centrality* awards the same score for every network neighbour, *eigenvector centrality* takes into account the importance of those neighbour nodes and treats them accordingly. So it does not only award one score point for each neighbour but it gives each node a score proportional to the sum of the scores of its neighbours (Newman, 2010: 169).

Eigenvector centrality can be calculated for both directed and undirected graphs but works best for undirected cases. This is due to the fact that an undirected graph has an asymmetric adjacency matrix. It has two sets of eigenvectors, but only one can define the centrality. The solution is normally to take the incoming connections as the important ones as in most contexts it is more important to “be voted” than to “vote a lot”. In the present corpus experiment this value therefore takes into account how many times a node is head for other nodes and the importance of those dependent nodes.

Nevertheless, *Eigenvector centrality* has a disadvantage with directed graphs which has to be taken into account. Nodes with many outgoing connections will still have a value of 0 and so be the same as a node with only one single outgoing connection. Furthermore these nodes with value 0 will then send their “0-value vote” to their heads and make only strongly connected components in the network being rewarded. *Katz centrality* and *PageRank* are alternative measures which address this problem.

Figure 8 exemplifies the same network as seen in Figure 7, but with the network neighbours of *casa* marked by a circle.

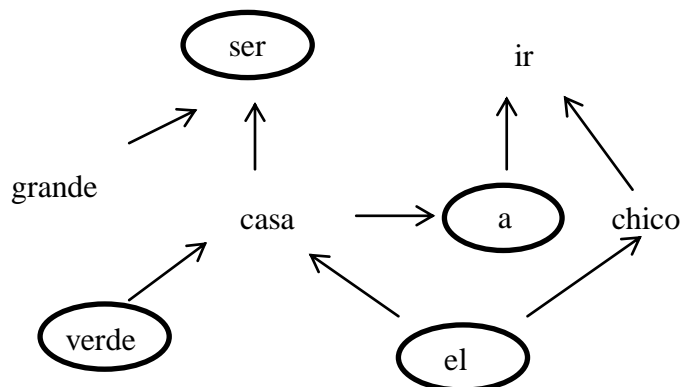


Figure 8: Example for network neighbours in a syntactic dependency network

- **Katz centrality**

This measure builds up on the *eigenvector* idea but adds a constant to the score which in this way solves the described “0-value” problem. This method was first proposed by Katz (1953). It allows nodes with many neighbours to be considered as important irrespective of those having a high centrality. The given importance can be adjusted by means of the constant. This can be a good solution for many network analyses but there is still a point left to speak about.

If one node has a high centrality and it points to many other nodes, it gives all of them a vote of high centrality. An issue for web pages that led up to the implementation of Google’s famous *PageRank* algorithm.

- **PageRank**

Google’s website classification for search results is based on this algorithm which adds an interesting feature to *Katz centrality*. Nodes with high centrality do not give anymore automatically high scores to all of its outgoing nodes but divides its vote score into small pieces according to the number of outgoing nodes.

This means in terms of websites that an important site which links to 100 other sites will give them each only one-hundredth part of the scores as if it only linked to one other page. The effect is that network hubs like Yahoo will not have a disproportionate influence on the search rankings.

In a dependency network this means that nodes with many different heads have to share their importance between all heads and do not give high votes to all of them. While this measure is adequate for web pages, it does not modify *eigenvector centrality* in a correct way for the analysis of a language network. In the latter, it is not a disadvantage for a node to connect to many heads and the used algorithm should not penalise this. Therefore it was decided to use *eigenvector centrality* for the final evaluation and not *PageRank*.

- **Closeness Centrality**

This measure gives information about the mean distance from a given node to the other nodes in the network. It calculates the shortest path between the pair of nodes in context and shows therefore small values for nodes which are separated only by a short distance. Afterwards the average value of all shortest paths between a specific node and all other nodes is calculated.

The importance of this value can be seen since a short path between two nodes normally implies a better and faster communication between them. It has to be taken into account that the results tend to span over a small range from largest to smallest which implies that an analysis has to consider changes of small amounts as significant.

In the final evaluation of this project, this value refers to the number of steps one node has to walk through the network along its heads in order to arrive at other nodes of the network. An optimised network should therefore show smaller numbers than one that is not.

- **Betweenness Centrality**

Betweenness centrality measures the degree to which a node lies on the shortest paths between other nodes of the network. Nodes with high *betweenness centrality* do not necessarily have high importance for the network in terms of former seen centrality measures as total-degree or closeness centrality but show their importance as “information deliverers”. Information travels through the network by help of these nodes and their removal can immediately have strong impacts on the network performance.

Their importance can therefore be seen in the flow of information through the network. The result can be presented as a total count of shortest paths which run through a specific node, which implies high numbers for big networks. Another option is to

provide a normalised value which divides the total count by the total number of (ordered) vertex pairs. In the latter case, the result will lie between 0 and 1.

In the final evaluation of this project *betweenness centrality* shows the importance of a lemma as “intermediary” between other nodes since it lies on the shortest path between them. Figure 9 shows the shortest path between *verde* and *ir* in the previously seen syntactic dependency network (Figures 7 and 8). Both *casa* and *a* lie on the shortest path and increase their betweenness centrality in this way.

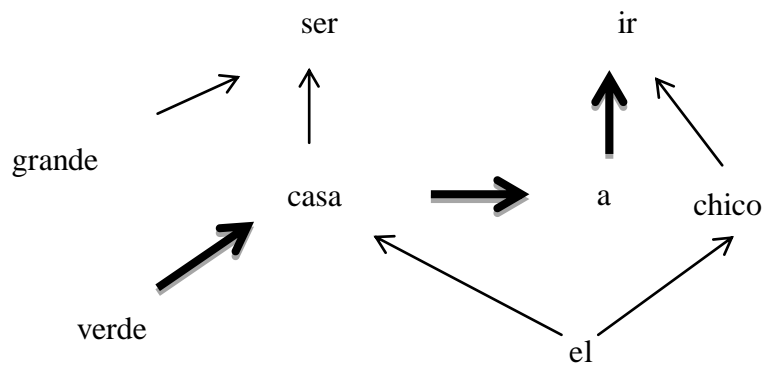


Figure 9: Example for *betweenness* in a syntactic dependency network

4. Surface Syntax Dependencies

The previous chapter has described already the original AnCora dependency annotation, and has also explained the problems it included for its use in this dissertation. Chapter 4 gives further details about the benefits of a new dependency annotation for this project and specifies the new annotation. This includes the motivation for the new annotation (Section 4.1), a look at related work regarding the AnCora dependencies (Section 4.2), a detailed explanation of the new dependency annotation for this project (Section 4.3)⁴ and the presentation of the deconstruction process of AnCora multiwords (Section 4.4).

4.1 Motivation

The idea is to have a base dependency annotation which is easily adaptable to NLP task-specific models. While most current approaches to dependency functions imply a semantic orientation, as for example the Stanford parser (De Marneffe and Manning, 2012) and also the AnCora Dependencies (Civit et al., 2006), it is preferable for the chosen approach in this project that the planned base version has a syntactic orientation in order to adapt it afterwards according to the needs of specific models.

The differences between a syntax-based and a semantics-based dependency annotation can be found in many linguistic structures. A noun phrase like *el resto de los chicos* ('the rest of the boys') can have *resto* ('rest') as its head under a syntactic point of view, while a semantic approach would chose *chicos* ('boys') as its head. Other differences between syntactic and semantic head selection can be found in other structures throughout the language such as verbal periphrases, modification relations, etc.

The available AnCora Dependencies (Civit et al., 2006) have a semantic orientation and are therefore not exactly what this project needs as base data. Nevertheless, as the corpus AnCora (Taulé et al., 2008) also offers a constituent annotation, the possibility

⁴ Furter details can be found in the article *From constituents to syntax-oriented dependencies* (Kolz et al., 2014a).

came up to create a new, syntax-based dependency annotation. AnCora Dependencies themselves were created by means of a semi-automatic conversion from AnCora constituents, therefore a similar approach with different linguistic criteria seems to be a feasible solution for the creation of surface-syntax dependencies, which will serve the project then as base data. Section 4.2 describes related work to automatic conversions of this type of data and gives information about further work done previously to this project on AnCora Dependencies. Section 4.3 explains the conversion process for the surface-syntax annotation and Section 4.4 the deconstruction of multiwords, which are present in AnCora Dependencies.

4.2 Related Work

The conversion from constituent to dependency structures is not new. Magerman (1994) made use of a head driven approach, which is still used and enhanced in newer works such as Collins (1999), Yamada and Matsumoto (2003) and Johansson and Nugues (2007). The approach has shown good results but there is still ongoing research.

As can be seen in such previous works, the resulting dependency tree structure depends highly on the focus of the annotation, which can apply either a syntactic or a semantic analysis. Johansson and Nugues (2007) mention the possibility to allow multiple-headed dependency structures to overcome this dichotomy.

In the particular case of the AnCora corpus, it is worth noting that its dependency relations annotation was carried out automatically by a conversion from constituents (Civit et al., 2006). Only a head and a function table were written manually. In many constructions, implicit semantic criteria are assumed in the linguistic decisions informing the conversion.

Along similar lines, Mille et al. (2009) present a reannotation of AnCora dependencies, already heading towards a more syntax-oriented approach. Their reannotation has been carried out semiautomatically and currently covers only a section of AnCora (100,892 out of 517,269 tokens). Their function tagset consists of 69 tags and is therefore quite

fine-grained for an automatic annotation. Given this and the fact that the resulting annotation is not yet available for the whole corpus, it was decided for this project to create an own tagset and proceed with an automatic annotation of the whole corpus. This process includes two individual tasks: a dependency relation annotation and a syntactic function labelling.

4.3 Annotation Description

The new dependency annotation is called Surface Syntax Dependencies (SSD) due to the linguistic decisions it is based on. It was important for this annotation that the taken decisions reflect a surface-syntax approach and that furthermore the automatic conversion implements these correctly. This section presents first of all the linguistic criteria (Subsection 4.3.1), which were established for this annotation. Then, the conversion process is described (Subsection 4.3.2) and the conversion results are presented (Subsection 4.3.3).

4.3.1 Linguistic Criteria

This subsection takes a closer look at the linguistic criteria that were adopted for grounding the dependency relations in this automatic conversion from constituents. First the decisions taken regarding the head selection of dependency relation are presented and afterwards the function labelling is explained with more details.

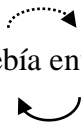
- **Dependency relations**

The goal of this annotation is to obtain pure syntax-oriented dependency structures. Thus, the chosen linguistic decisions are compliant to that.

➤ Periphrastic verbs

In this annotation, auxiliary and modal verbs are the head of the structure, as shown below in (41) and (42). In the following examples, the upper graph shows the original AnCora treatment and the lower one the surface-syntax-based decision.⁵

(41) debía enviar los rollos



he had to send the reels

(42) ella ha estado viendo la exposición

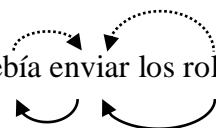


she has been seeing the exhibition

The semantic criteria applied by AnCora converts the main verb into the head, while the conjugated auxiliary verb is a dependent of former.

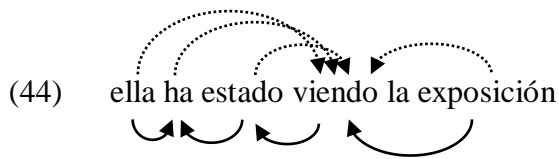
As the auxiliary verb is in agreement to the subject, the syntax-based annotation should make subjects depend on the auxiliary or modal (as marked by the agreement relation) and other complements, on the main verb. This treatment is exemplified in (43) and (44).

(43) debía enviar los rollos



he had to send the reels

⁵ The arrow points at the head of the dependency relation



she has been seeing the exhibition

➤ Complex nominal phrases

The treatment of complex nominal phrases like *el resto de los chicos* (‘the rest of the boys’) illustrates the differences between a semantic and a syntactic approach. In Spanish the agreement between the nominal phrase in (45) and the related verb can be based both on *resto* (‘rest’) and on *chicos* (‘boys’). While a semantic approach would chose *boys* as head, a syntax-based annotation should prefer *resto* in this place. AnCora always chooses the semantic head, while Surface-Syntax Dependencies prefers the syntactic head.

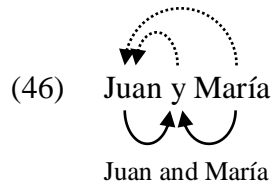


the rest of the boys

➤ Coordinations

A coordination structure contains at least two elements which are coordinated by one or more conjunctions. Head candidates are one of the coordinated items or one of the conjunctions. AnCora sees the first coordinated element as head, while the SSD model identifies as head the conjunction. The reason behind this can be easily seen in example

(46) since the nominal phrase would take a verb in plural but both coordinated elements inside are singular forms. The agreement follows out of the joined result of both. Therefore a syntax-based annotation should not take one of the coordinated elements as head but the conjunction.



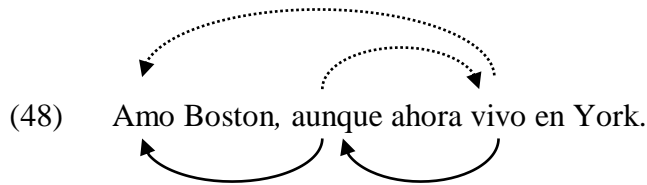
In case of coordinations with paired conjunctions (e.g., *ni...ni...*, ‘neither...nor...’), the last conjunction is treated as the head of both the conjuncts and any former conjunction or comma (47).



This approach has the advantage that all coordinated elements depend on the same node and can be found at the same level within the dependency tree.

➤ **Subordinating conjunctions**

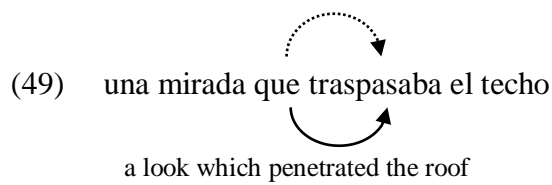
The conjunction is the head of the subordinated clause, in full accordance to the surface syntactic structure. By contrast, AnCora identifies the verb of the subordinated clause as head and sees the conjunct as its dependent, as shown in (48).



I love Boston, although I now live in York.

➤ Relative clauses

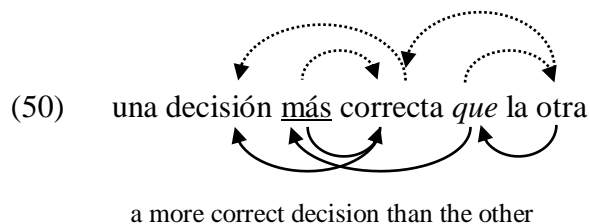
The verb of the relative clause is also its head, while the relative pronoun is its dependent. This case has been treated differently than other subordinating structures given the double role of the relative pronoun (as connector and as argument of the main predicate in the subordinated clause). An example can be seen in (49).



This analysis corresponds to the same treatment as seen in AnCora.

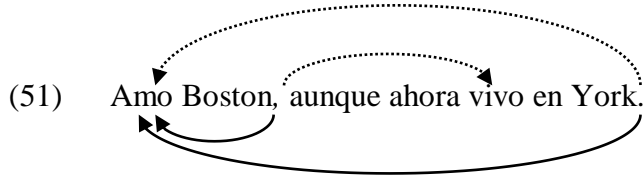
➤ Comparative Structures

The comparative element (e.g., *más*, ‘more’) depends on the adjective (*correcta*, ‘correct’) and at the same time is the head of the embedded phrase (*que la otra*, ‘than the other’). The embedded phrase sees the conjunction (*que*) as its head. In AnCora, the conjunction of the embedded phrase did not connect it to the upper part of the comparative structure.

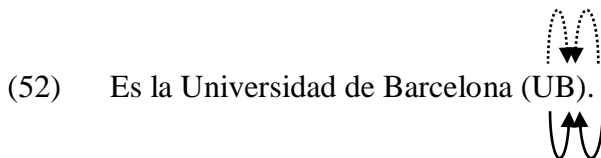


➤ Punctuation

Commas and full stops are seen as dependent of the higher constituent head (51). In AnCora dependencies the commas are dependents of the lower constituent head. Brackets, quotation marks, etc. are seen as dependent of the head within their constituent range (52). This is the same treatment as in the original AnCora annotation.



I love Boston, although I now live in York.



It's the University of Barcelona (UB).

• Function Tagset

The syntactic functions tagset has to fulfil two requirements. It has to be as informative as possible and must be of reasonable size in order to guarantee a successful automatic annotation. The word *reasonable* refers in this case to a quantity which is adequate for a high quality automatic annotation and which at the same time provides enough linguistic information to be useful in further NLP processing.

The tagset used in AnCora has 51 tags, thus being of a reasonable size. However, it has the problem of mixing dependency relations with part-of-speech and constituent structure tags. Some examples:

- **Dependency function tags:** suj (subject), cd (direct object), ci (indirect object).
- **Constituent structure tags:** sn (nominal phrase), s.a (adjectival phrase).
- **Part-of-speech tags:** v (verb), n (noun).

On the other hand, the Stanford tagset (de Marneffe and Manning, 2012) seems to be adequate for both requirements. The size of 52 tags is reasonable for an automatic annotation and the individual tags are a good choice to represent dependency relations information. In addition, tags are structured in a hierarchical way, thus allowing underspecified tags when required. In the present proposal, Stanford’s tagset is adapted for Spanish (e.g., *reflec*, reflexive) and enhanced with some tags already available in AnCora (e.g., *te*, textual element) in order to increase its informativeness.

Table 8 shows a list of Stanford tags which are not included in the tagset for SSD. This is due to different reasons, some tags were too specific to be considered in this approach (such as *discourse*, *quantmod*, etc.), others refer to a semantic point of view (such as *aux*, *mwe*, etc.) and some tags are not relevant to Spanish (*possessive* and *pri*).

acomp	mwe	predet
aux	npadvmod	prepc
auxpass	nsubjpass	pri
cop	number	quantmod
csubjpass	parataxis	ref
discourse	pcomp	vmod
expl	possessive	xcomp
goeswith	preconjunct	xsubj

Table 8: Not considered Stanford tags

There are also two tags which are only renamed, but used in the same sense as in Stanford Dependencies (Table 9). In this way, they have a name which is closer to the Spanish naming of the syntactic structure.

Stanford tag name	SSD tag name
cc	coord
ccomp	cobj

Table 9: Renamed Stanford tags

Two tags were directly taken from the AnCora annotation as they add useful information to the annotation and are already implemented in the AnCora constituents annotation, which serves as input for the later conversion (Table 10).

te	cpred
----	-------

Table 10: Tags added from AnCora dependencies tagset

Furthermore, 12 tags were added to the SSD tagset (Table 11) in order to provide expressiveness to the surface-syntax approach and give related specific information when needed (f. ex. *prepv*, *prepn* and *prepa* specify the part-of-speech of a preposition’s head).

cobj	prepa
cobj	arg
oobj	attr
vobj	infmod
prepv	partmod
prepn	reflec

Table 11: Tags added to SSD dependencies tagset

The SSD tagset is presented in Table 12. It contains 42 function tags (including underspecified ones), which makes it fully adequate for an automatic annotation (Subsection 4.3.2).

In the table, indentation shows the tagset hierarchical structure, conveying that general tags like *obj* or *mod* include more specific subclasses. In the annotation, the goal is obviously to be as specific as possible, since this leads to more informative data. Therefore, the generic tags like *dep*, *comp*, *obj*, *mod* and *prep* are not expected to be of common use but only for cases where a more specific tag cannot be applied.

Tag	Full name
root	root
dep	dependent
arg	argument
comp	complement
attr	attributive
cpred	predicative complement
obj	object
cobj	complementizer object
dobj	direct object
iobj	indirect object
oobj	oblique object
pobj	object of a preposition
vobj	object of verb
crobj	object of comparative
subj	subject
nsubj	nominal subject
csbj	clausal subject
coord	coordination
conj	conjunct
agent	agent
reflec	reflexive (“se”)
te	textual element
mod	modifier
abbrev	abbreviation modifier
amod	adjectival modifier
appos	appositional modifier
advcl	adverbial clause modifier
det	determiner
infmod	infinitival modifier
partmod	participial modifier
advmod	adverbial modifier
neg	negation modifier
rcmod	relative clause modifier
nn	noun compound modifier
tmod	temporal modifier
num	numeric modifier
prep	prepositional modifier
prepv	prep. mod. of a verb
prepn	prep. mod. of a noun
prepa	prep. mod. of adjective
poss	possession modifier
punct	punctuation

Table 12: Surface-Syntax Dependencies function tagset

4.3.2 Automatic Conversion

The conversion from constituent structures to dependencies is only done for the Spanish part of the AnCora corpus (Taulé et al., 2008). The AnCora constituent annotation contains 17,376 sentences which are split over 1,636 files and gathers a total count of 517,269 tokens. The data is annotated for different linguistic levels, both constituent structures and dependency relations included (see Subsection 3.1.1). All sentences are tokenized, and tokens have information on their lemma and part-of-speech.

- **Process**

The system takes the constituent structure layer in AnCora as input and builds the syntax-oriented dependency trees supported by linguistic rules.

The core of the process is identifying the head of each constituent, along the lines of Magerman (1994) and subsequent work. The dependent nodes can then be pointed to the identified head. One single main rule selects the head in all clearly headed constituents in the corpus. However, a remarkable number of constituent structures in AnCora are not clearly headed, because they are flat structures or conflate several nodes into one (e.g. the verbal group formed by the main verb and its auxiliaries or modals). To tackle these cases a set of nine finer grained rules are added (two for flat constructions and seven for divergence in head selection).

Once the dependency structures are obtained, the syntactic function of each head-dependent pair is determined. The function labelling process is informed with data from two sources: the part-of-speech of both nodes in each pair, and the argument-structure function tags that had been manually annotated in the AnCora constituent structure layer (subject, direct and indirect object, oblique and textual element). Based on those two elements, rules can be established to automatically annotate the syntactic functions between head and dependent node.

- **Algorithm**

The applied algorithm is as shown in Figure 10.

```
1 function DEPENDENCY_ANNOTATION(parsed_text):
2   for sentence in constituents:
3     read_constituents_tree(sentence)
4     for constituent in constituents_tree:
5       identify_head_of_constituent(constituent)
6       # uses a preference list for possible candidates
7
8     for terminal_node in constituents_tree:
9       walk_constituents_tree(terminal_node)
10      # bottom-up
11      # walks tree until not head anymore and
12      # connects there as dependent to head
13     for terminal_node in constituents_tree:
14       label_functions()
```

Figure 10: Conversion Algorithm

The procedure takes the parsed text as input (line 1), analyses it sentence by sentence (line 2) and generates its dependency structures. In particular, the program reads the constituent tree of each sentence (line 3) and identifies the head of each constituent (line 5). The procedure then walks bottom-up from terminal nodes through the constituent structure and connects them to their head (line 8). Finally, each relation between dependent and head is labelled according to the function tagset presented in Table 12 (line 13).

- **Issues**

The conversion from constituent structures to dependency structures is highly dependent on the input that comes from the constituents. Thus, inconsistencies in the constituent annotation may lead to problems when applying the general procedure.

Three specific issues were encountered: grouping of several lexical items as a single token (e.g., *la_mayoría_de*, ‘the majority of’), in Ancora referred to as multiword, the depth of annotation in constituent trees (e.g., *debía haberlo resuelto*, ‘should have solved it’, as a flat structure), and the presence of empty tokens signalling subject ellipses.

➤ Flat structures

Flat structures posed a problem for the identification of heads and their dependents since they often contain several constituent heads: the head of the constituent and another head of what should have been a lower constituent, as underlined in (53).

(53) S=conj S grup.verb sa sn sp

In this example, a deeper analysis is expected which groups together also *grup.verb sa sn sp* to an S.

This problem is tackled by specific rules which detect flat structures and insert an intermediate structure introducing the different heads and their corresponding dependents. This way they can be treated as well-formed constituents.

➤ Multiwords

In AnCora, multiwords include complex prepositions or conjunctions, verb groups, complex determiners and proper names. They are challenging because many of them are treated sometimes compositionally and sometimes as a single token:

(54) a. ya_que
b. ya que

This multiword approach is not adequate for a purely syntax-based analysis and therefore it was decided to deconstruct them. The deconstruction process into individual tokens is explained with all details in Section 4.4.

➤ **Empty elements**

Another modification to the original AnCora annotation is the suppression of empty tokens which correspond to dropped subjects in Spanish. Since these items do not appear in the text, it seems preferable for the proposed syntax-based annotation to exclude them in the dependency tree.

4.3.3 Evaluation

The following lines explain how the conversion has been evaluated and include a more detailed look on the created evaluation corpus and on the error analysis in order to show what kind of problems came up in the conversion process.

- **Evaluation Corpus**

The evaluation corpus was annotated manually for both dependency relations and syntactic functions. A total of 256 sentences were annotated which were chosen partially randomly; that is, it was made sure that the selected files included all linguistic phenomena described in Subsection 4.3.1. The evaluation corpus contains a total of 6,160 tokens (out of the 517,269 tokens in AnCora, which corresponds to a 1.5 % of the of the files in the whole corpus).

	Sentences	Tokens	Files
AnCora Dependencies	17,376	517,269	1,635
Evaluation corpus	256	6,169	21

Table 13: Evaluation corpus size overview

The annotators were two professional linguists, which annotated each half of the evaluation corpus. Additionally, complex structures were discussed between both of them.

Figure 11 exemplifies the content and format of the evaluation corpus:

1#La #2#det
2#situación #10#nsubj
3#en #2#prepn
4#las #5#det
5#carreteras #6#coord
6#y #3#pobj
7#las #8#det
8#montañas #6#coord
9#se #10#reflec
10#normalizó #ROOT#root
11#en #10#prepv
12#todas #14#det
13#las #14#det
14#autonomías #11#pobj
15#afectadas #14#amod
16#. #10#punct

Figure 11: Evaluation corpus fragment

- **Results**

The obtained results show a labelled attachment score (LAS) of 0.85, an unlabelled attachment score (UAS) of 0.92 and a label accuracy (LA) of 0.89.

	Accuracy	Kappa
LAS	0.85	-
UAS	0.92	-
LA	0.89	0.88

Table 14: SSD Conversion Results

Since syntactic function labels are likely to get an incorrect result if the corresponding node's head was not set correctly, we also calculated the label accuracy of the correctly identified attachments, which was 0.93.

The Kappa coefficient K for agreement between coders has been calculated in order to exclude the factor of agreement by chance. Among the two main ways of calculating Kappa we followed Cohen (1960) because it is better suited for cases where categories have significantly different distributions. In this case the coders were a human annotator and our system. The kappa value for syntactic function labels of 0.88 is in the range of almost perfect agreement according to Landis and Koch (1977).

Unfortunately, Civit et al. (2006) do not give results for their conversion from constituents to dependencies in their paper. These results would have been the best comparison for our results as they are based on the same corpus even if not tagged with the same function tagset.

- **Error Analysis**

The error analysis splits into errors observed in the dependency relation identification task and errors in the labelling of the relation.

- **The dependency tree creation**

The observed data shows that the system had problems with complex coordinated structures as, for example, citations which contain more than one sentence.

(55) He said: "Sentence 1. Sentence 2"

In addition, the rules which treated flat constituent structures were not always able to create the correct dependencies for deeper nodes.

➤ **Function labelling**

The results and exact frequencies of agreement and disagreement between the manual annotation and the system's one are presented in a confusion matrix (Appendix A) which counts only the labels of correctly related dependencies.

As the matrix shows, the system had problems with some coordination structures. 72 out of 348 cases showed an incorrect label. Problems came up especially in cases of complex structures, particularly with correlative conjunctions (like *bien... bien...*, 'either... or...').

In other cases the rules were too generic. An example can be seen in the labelling of the function *attr*. The system looks at the head lemma and sets *attr* if it is *ser* ('to be'). Cases were found in which the label was wrongly used in passive contexts like *han sido absueltos* ('they were absolved'). The confusion matrix shows that in 10 out of 64 cases the system wrongly identifies the function as being *attr* instead of *vobj*.

Such generic cases could be improved by rewriting some rules in a more specific way. Furthermore, the system does not include rules for the use of generic labels like *obj*. Thus it always assigns a specific label and if this does not fit, it currently assigns the label *dep*.

Some not so frequently used labels like *nn* or *abbrev* could not be tested as they did not appear within the evaluation corpus.

4.4 Multiword Deconstruction

The following section describes some basic ideas regarding the concept multiword and explains afterwards the deconstruction process applied to the AnCora corpus.

4.4.1 Multiword Theory and AnCora Treatment

Before the deconstruction process itself is explained, some basic ideas about the concept *multiword* and its treatment in AnCora are explained.

4.4.1.1 Multiword Definition

A multiword can linguistically be described as "idiosyncratic interpretations that cross word boundaries (or spaces)" (Sag et al., 2002: 2). In AnCora, a multiword groups them together in one single token (by means of underscore characters). Its usage can be found in almost all types of part-of-speech (Figure 12).

Universitat_de_Barcelona (proper noun)
página_web (common noun)
querer_decir (verb)
on_line (adjective)
de_nuevo (adverb)
a_pesar_del (preposition)
cien_mil (number)

Figure 12: Multiword part-of-speech examples

Furthermore, it is noticeable that time expressions (e. g. *2_de_marzo_de_1995*) fall within the applied multiword concept and that one can find even complex structures within multiwords such as coordinations (e. g. *Industria_y_Comercio*) or nouns modified by adjectives (e. g. *el_bucle_melancólico*).

4.4.1.2 Internal Structure of Multiwords

A multiword contains at least two tokens with theoretically no limitation on the number of maximum tokens. AnCora joins tokens to a multiword by making use of the

underscore character (e. g. *Universitat_de_Barcelona*). As a multiword only occupies one token in the AnCora annotation, it has also only one syntactic function assigned (Figure 13).

17	presentada	presentado	15	S	
18	en	en	17	cc	
19	la	el	20	spec	
20	Universitat_de_Barcelona	Universitat_de_Barcelona	n	18	sn

Figure 13: AnCora multiword example (5th column: syntactic function)

The internal head of the multiword is dependent of a node outside of the multiword. In the example of Figure 12 *Universitat* ('University') would be the internal head of the multiword structure and it would be dependent of *en*.

The head-dependent relations among tokens within the multiword are not expressed. This information will have to be calculated in a deconstruction process as in principle any token within the multiword is a candidate for being the head of a head-dependent pair.

This is actually the case as multiwords can contain complex structures such as coordinations or specifiers. These structures inside the multiword have to be analyzed according to the criteria set up for the annotation and each token needs to get attached to a head within the multiword range. The part-of-speech of the tokens can give valuable information for the identification of their head and the setting of the syntactic function.

4.4.1.3 External Relations

A multiword has on the one hand a head (upper head) of which it is a dependent and on the other hand it can be head of further nodes (lower dependency relations). Figure 14 exemplifies this constellation. Identifying the head of the internal multiword head in a deconstruction process is straightforward as it keeps this relation from the multiword (attached to the upper head).

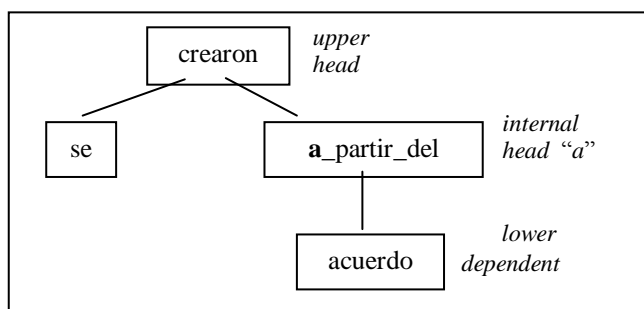


Figure 14: Example for upper head, internal head and lower dependent

The correct setting of lower dependents of the former multiword is more complicated as there is not a simple default solution. Figure 15 shows different examples of multiwords and their dependents.

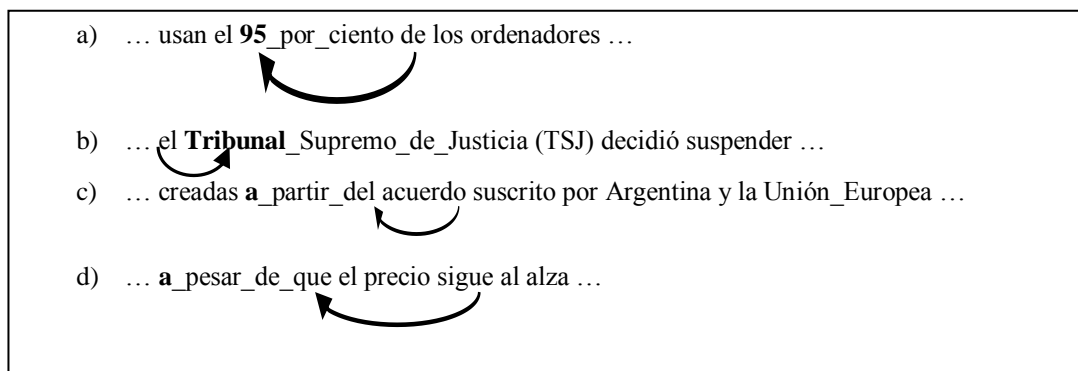


Figure 15: Examples of multiwords and their variety of dependents

Again the part-of-speech gives us important information of how to set up the relation to lower dependents. Normally, a lower dependent will connect to the head of the multiword structure (a and b in Figure 15). But a preposition or conjunction in the last position of the multiword will lead to a different treatment as they work as head for lower dependents (c and d in Figure 15).

4.4.2 Multiword Deconstruction Process

The following subsections give further details about the multiword deconstruction.

4.4.2.1 Motivation

The deconstruction of AnCora multiwords was necessary as the concept of multiword tokens is not compatible to a surface-syntax approach. Additionally, the AnCora corpus showed certain inconsistencies in their treatment. A few examples can be found in table 15.

Token use in AnCora	Multiword use in AnCora
aún así	aún_así
mayo pasado	mayo_pasado
de hecho	de_hecho
mientras que	mientras_que

Table 15: Example of inconsistencies in AnCora multiword treatment

A parser would have problems with this data as it would have to be aware of the possibility to find a multiword written together as one token but also as a sequence of tokens. The same happens with searches over the corpus, if one is interested in gathering all temporal expressions, for example, it has to be considered that they can be found within multiword tokens but also outside of them.

Furthermore, the concept of writing a group of words together as if it were one word is not expressed in natural language, and introduces a source of artificiality over the word forms of the corpus. These points emphasise the decision to deconstruct the multiwords.

4.4.2.2 Multiword Statistics

AnCora contains a total of 9,113 types of multiwords which make up 18,953 instances in the whole corpus. The multiwords have a wide range of lengths, the majority are two-token multiwords but the longest entry is an 18-token multiword. Table 16 shows the distribution of multiword instances according to their token length. The upper row

indicates the length of the multiword in tokens and the lower one shows the number of instances found with the corresponding length.

2	3	4	5	6	7	8	9	10	11	12	13	18
11,520	5,454	1,144	459	188	79	59	27	12	5	3	2	1

Table 16: Multiword lengths statistics

4.4.2.3 Algorithm

An overview of the multiword deconstruction algorithm is presented in Figure 16 and further explanations may refer to the indicated line numbers. The program starts the deconstruction process by reading first all AnCora multiwords and storing their types (line 2). All tokens which can be found within multiwords get then labelled with their part-of-speech (line 3) and afterwards a part-of-speech sequence table is created which gathers possible deconstruction settings for multiwords based on their part-of-speech combination (line 4). These possible solutions come from multiwords which were also treated as separated tokens in AnCora and also by the creation of further token combinations which correspond to needed part-of-speech sequences and which are connected among each other in head-dependent relations.

```

1 Function Multiword_Deconstruction(dependencies):
2     multiwords=get_multiwords(dependencies)
3     add_pos_to_multiwords(multiwords)
4     pos_sequence_table=create_pos_sequence_table(dependencies)
5     classifier=classify_multiwords(multiwords,pos_sequence_table)
6     for sentence in dependencies:
7         deconstruct_multiwords(sentence,classifier)

```

Figure 16: Multiword deconstruction algorithm.

Afterwards, a classifier takes all multiwords and sets all head-dependent pairs within the multiword and their respective syntactic function according to the solution proposed by the part-of-speech sequence table.

Boca_Júniors	:[0, 1],['dobj', 'appos']
El_Noticiero_Universal	:[2, 0, 2],['det', 'coord', 'amod']
17_de_octubre	:[0, 1, 2],['dobj', 'prep', 'pobj']

Figure 17: Multiword classifier output.

As one can see in Figure 17, each token gets a head within the multiword (besides the one with the value 0 which is in this way identified as internal head of the multiword structure) and a syntactic function label according to its dependent-head relation. While the syntactic function labels of the internal multiword tokens generally stay the same in all kind of contexts, it is worth to comment that the label assigned to the relation of the internal head of the multiword to its upper head varies according to the context as it depends on the syntactic configuration of each case. So, the first entry in the label list of *Boca_Júniors* could as well be a *nsubj* if it was used as a subject in a certain sentence. This label is therefore set in the deconstruction process. In case that the classifier could not find a solution for the part-of-speech combination of the multiword, a default rule was set up which connects all tokens in the multiword from right to left according to their position, only taking into account the treatment of determiners and adjectives modifying nouns. This means that in the deconstruction of a multiword like *Jurado_Nacional_de_Elecciones* the preposition *de* (‘of’) would still be attached to *Jurado* (‘jury’) and not to *Nacional* (‘national’), even if the classifier uses the default rule.

Finally, each sentence of the corpus is passed through the program, which by help of the classifier deconstructs all multiwords setting the head of each multiword token and its respective syntactic function. The deconstruction process handles by rules the treatment of lower dependents of multiwords. AnCora Surface Syntax Dependencies are then annotated. Figure 18 shows an example for a deconstructed multiword annotation.

1	El	3	det	da0ms0	el	
2	ex	3	amod	aq0cn0	ex	
3	ministro	12	nsubj	ncms000		ministro
4	español	3	amod	aq0ms0	español	
5	de	3	prepn	sps00	de	
6	Industria					
7	y	5	pobj	cc	y	
8	Energía	7	coord	np00000		Energía

Figure 18: Example of a deconstructed multiword with a coordination inside

4.4.3 Evaluation of Multiword Deconstruction

AnCora contains 18,953 multiword instances which correspond to 9,113 multiword types. Since multiwords were classified in this approach by type, the evaluation was based on an amount of 500 multiword types. This makes up around a 5.5 % of the total amount.

Since the classifier of the program does not include solutions for all part-of-speech combinations found in multiwords, the evaluation has to consider this by taking into account a corresponding amount of multiwords which were deconstructed by the default rule. 459 of the 9,113 multiword types were classified in this way, so it was decided to include a 5 % (25 of 500) of those default solutions also in the evaluation in order to get meaningful results.

The evaluation consisted then in a manual revision of 500 selected multiword types by checking all their individual heads and their syntactic function label. Those 500 multiwords contained a total of 1,374 tokens. The results (Table 17) obtained are highly satisfactory as label accuracy (LA) reached 0.92, the unlabeled attachment score (UAS) 0.96 and the labelled attachment score (LAS) a value of 0.92. The fact that both LA and LAS show the same result can be explained as the setting of the syntactic function label is highly dependent on a previous correct identification of its head and the overall high accuracy results.

	Accuracy
LA	0.92
UAS	0.96
LAS	0.92

Table 17: Multiword deconstruction evaluation.

4.4.4 Final Version of AnCora Surface Syntax Dependencies

The final version of AnCora Surface Syntax Dependencies contains 547,724 tokens. The change in token count compared to the former AnCora dependency annotation (517,269 tokens) results from the deletion of elliptic subjects and the deconstruction of multiwords.

The following table shows the distribution of the tagset in AnCora Surface-Syntax Dependencies. The label *dep* was set if the system could not identify a more detailed label and this was the case in only around a 1.5 % of the corpus. It was also checked that each sentence had a *root* node and not more than one *root* as this a requirement for a correctly parsed sentence.

Tag Name	Frequency
pobj	87,409
det	71,332
punct	65,282
prepn	42,021
dobj	32,534
coord	30,305
amod	29,713
nsubj	29,316
prepv	22,405
root	17,361
cobj	16,848
advmod	16,049
appos	15,631
rmod	7,820
dep	7,421
attr	6,814

poss	5,501
reflec	5,313
oobj	5,133
vobj	4,363
advcl	3,944
neg	3,923
iobj	3,535
prep	3,463
prepa	3,453
num	3,352
cpred	2,236
conj	1,497
agent	1,411
tmod	814
te	783
csbj	395
crobj	329
voc	13
mod	3
partmod	2

Table 18: SSD tagset sorted by frequency

It should be commented at this point that the difference between the number of *root* tags (17,361) and the official number of sentences in AnCora (17,376) came up as a result of empty sentences in AnCora which were left out in the new annotation. See Figure 19 for an example of an empty sentence in the constituent annotation and Figure 20 for its correspondence in AnCora dependencies.

```

<sentence title="yes">
  <sn entity="entity2" missing="yes" title="yes"/>
</sentence>
<sentence>

```

Figure 19: Empty sentence in AnCora constituents (file: CESS-CAST-P_107_20010802.txt)

```
1 _ _ _ 0 sentence sn _
```

Figure 20: Empty sentence in AnCora dependencies. (file: CESS-CAST-P_107_20010802.txt)

5. Temporal Dependencies

As seen in the previous chapter, the project has now a syntax-oriented dependency corpus available which can be used as the basis for the adaptation to task-based dependency models. Chapter 5 presents the first task-based dependency model, which is called Temporal Dependencies and dedicated to the NLP task of temporal parsing.

First, the chapter describes how time information is conveyed in Spanish (Section 5.1). It is important to understand which possibilities exist to express time information, since the dependency annotation of the model shall facilitate temporal parsing purposes in NLP applications. Based on the observations made, an approach for the model creation is taken (Subsection 5.2.1). Afterwards, the temporal dependency tagset is presented (Subsection 5.2.2), which is enriched with time information and adapted to temporal parsing purposes, and the implementation is discussed in detail (Subsection 5.2.3). Finally, Subsection 5.2.4 analyses the results of the process for adapting surface-syntax dependencies to this new model.

5.1 Ways to Express Time in Spanish

When humans talk about events and situations, they normally give the hearer (or reader) information about the corresponding time. This temporal information can be vague or exact and be transmitted by different ways in natural language.

On the one hand, speakers can make use of the whole inventory of time expressions, which the vocabulary of a certain language (in this case Spanish) offers, and, on the other hand, grammatical instruments like verbal tenses are used to indicate at least the direction of the event on a temporal axis.

Temporal information is conveyed in Spanish by means of lexical markers and grammar. The lexical markers can be divided into two types: *anchoring* and *ordering*. While anchoring temporal markers place an event to a position on the timeline, an ordering temporal marker temporally orders with respect to another event or time point.

The following subsections will take a closer look on the ways to express time in Spanish. They are divided into one subsection about lexical markers (5.1.1) and one about grammar (5.1.2).

5.1.1 Lexical Markers

The lexical markers can be divided into two groups: anchoring and ordering lexical markers. The anchoring lexical markers belong to the category *time expressions* and gather different types of part-of-speech. The ordering lexical markers make also use of different parts-of-speech, such as adverbs and conjunctions, but cannot be attributed to a fixed concept as the previous mentioned time expressions.

5.1.1.1 Anchoring Time Expressions

Anchoring time expressions give the reader information about the position of an event in the timeline. It is important for the planning of this task-based dependency model to know about the types of time expressions, which can be found, and their internal structure.

- **Types of Time Expressions**

A speaker introduces intentionally time expressions into his utterance to refer it explicitly to a certain time point or time span on the timeline. This can be done by giving an exact calendar date like in (56).

(56) La reunión tuvo lugar **el 12 de diciembre de 2011**.

The meeting took place on 12th of December of 2011.

But also by giving a not wholly specified time expression which is implicitly completed by the context, like (57).

(57) **El lunes** iré a Sevilla.

On Monday, I will go to Seville.

As the time expression in (57) needs another fully specified time point in order to refer to a concrete time point, this type of temporal expressions is referred to as *anchored temporal expressions* (Saurí, 2010). This means that the time expression in (57) anchors the event *iré* ('I will go') by using information from another time reference in order to be fully specified.

Nevertheless, not all time expressions refer to a calendar date. They can also refer to time spans like (58) or (59).

(58) Trabajó **tres meses** en la empresa de su padre.

He worked three month in his father's company.

(59) Se quedó **un rato** esperando.

He waited for a while.

The difference between (58) and (59) is that (58) contains an exact indication of a time span, while (59) includes a vague time span. One can therefore refer to (58) as a precise duration, while (59) is an example of a fuzzy one.

Durations in time can also be indicated by a start point and by an end point, so instead of giving the length of the duration like (58) one could also refer to that time by giving the months in which the period starts and when it ends like in (60).

(60) Trabajó **desde febrero hasta mayo** en la empresa de su padre.

He worked from February to May in his father's company.

This concept of indication of a duration by giving the starting and the end points can also be called a range. The indicated data in (60) is more concrete than (58), since (58) only indicates the length of the time span but not where to anchor it on a timeline.

Additionally, one can also indicate how often something happened, as in (61). These indications are called sets.

(61) **Dos veces por semana** voy al gimnasio.

I go to the gym two times a week.

This subsection has shown the variety of time expressions in Spanish; their identification in text is crucial for temporal parsing and a task-based dependency model should take this into account. An improved positioning of the relevant nodes in the dependency structure and their explicit identification can provide rich information to a temporal parser. The next subsection takes a look at the internal structure of time expressions.

• **Internal Structure of Time Expressions**

Time expressions can be found within a variety of parts-of-speech and in different dependency structures.

The part-of-speech of time expressions can be an adverb (such as *recién*, ‘recently’), an adjective (such as *próximo*, ‘next’), a noun (such as *enero*, ‘January’), a proper name (e.g. *Navidad*) and also numerals (1976, 19.47h, etc.).

Time expressions can also span along various words. In this case a time expression will have an internal head in the dependency analysis. In an example like *el próximo jueves* (‘the next Thursday’), the noun *jueves* would be the internal head of the time expression.

It has also to be taken into account that time expressions can be modified by words which do not belong themselves to the time expression (e. g. *un bonito domingo*, ‘a nice

Sunday’) or be connected to the rest of the sentence by means of a preposition (e. g. *para el mes de septiembre*, ‘for the month of September’).

The individual tokens which compose a time expression, and their modifiers, are already connected in SSD. The enhancement of relevant time information can therefore be applied to the head node of the time expression. An improved position of the head node will automatically lead to a better positioning of the whole time expression.

5.1.1.2 Ordering Lexical Markers

Spanish speakers have an inventory of lexical markers at their disposal in order to express explicitly the temporal relation of one event to another. The relation to the time location of the other event can be expressed in terms of *anteriority*, *simultaneity* and *posteriority* (Bosque and Demonte, 1999: 2876 ff).

The ordering lexical markers can thus be divided into those referring to anteriority (e. g. *antes*, ‘before’), to simultaneity (e. g. *al mismo tiempo*, ‘at the same time’) and to posteriority (such as *después*, ‘afterwards’). The part-of-speech found in this vocabulary shows variety as it includes adjectives (e. g. *anterior*, ‘previous’), ordinals (e. g. *segundo*, ‘second’), adverbs (e. g. *después*, ‘afterwards’) and conjunctions (such as *cuando*, ‘when’).

It should be taken into account that these lexical markers can span over various words and therefore have an internal head of the structure. The example *al mismo tiempo* (‘at the same time’) shows a prepositional phrase with the preposition *al* as internal head of the structure.

The treatment for these lexical markers in a task-based dependency model should be the same as the one commented in Subsection 5.1.1.1 for anchoring time expressions.

5.1.2 Grammar

Grammar can reflect anchoring and ordering of an event by the used verbal tense used. But, at least in Spanish, it is not just that a past tense refers to the past, a present tense to the present and the future tense to the future. Other important observations can be found by an analysis of grammar and its possible effects on time information. In particular, the treatment of modality, aspect and mood plays an important role since uncertain events cannot be matched on the same event timeline as certain ones (Subsections 5.1.2.2 and forward). The following observations were made by help of two Spanish grammars (Bosque and Demonte, 1999; Vera-Morales, 2004).

5.1.2.1 Tense

This section takes a closer look at the present, past and future tenses used in Spanish which can reflect the temporal anchoring and ordering of events.

- **Present tense**

First of all, the possibility has to be mentioned to refer to the past by a past tense form, to the present by the present tense and to the future by the corresponding tense.

Nevertheless, this is not all that can be observed. Spanish can use the present tense in some cases to refer actually to the future, as in (62). In this case an explicit lexical marker indicates this future reference.

- (62) **Se casan** el año que viene.
They get married next year.

But the present tense can not only refer to the present and the future, but eventually also include a moment in the past as can be seen in (63). The Spanish present tense is used

together with a temporal expression and refers to a duration that starts in the past and goes on until the present moment. Again, an explicit lexical marker is found to indicate this reference to the past.

(63) Desde hace meses **nos vemos** todos los lunes.

For the last month we have seen us every Monday.

Colloquially, the present tense could also be used to refer to an event that happened in the past as in (64).

(64) Casi me **caigo**.

(lit.) I almost fall down.

The present tense finds usage in the historical present (*presente histórico*) for the narration of past events, as in (65). A time expression is again included.

(65) Colon **descubre** América en 1492.

Columbus discovered America in 1492.

- **Past tense**

Spanish uses four different past tenses: *perfecto*, *indefinido*, *imperfecto* and *pluscuamperfecto*. Each tense has a different usage and implies temporal information besides the reference to the past.

➤ **Perfecto**

The *perfecto* is used for events that are closely related to the present. This can be because of the event happening temporally close to the present moment (66) or the effect of the in the past performed event is still active in the present (67).

- (66) - ¿**Has dicho** algo?
 - Have you said something?
 - No, no **he dicho** nada.
 - No, I haven't said anything.
- (67) - Las calles están mojadas.
 -The streets are wet.
 - Sí, **ha llovido** toda la noche.
 -Yes, it has rained the whole night.

The event *llover* ('rain') in (67) is still showing its effects in the moment of the utterance and makes it possible to use the perfect (*perfecto*). The auxiliary verb in the composed verb form is in this way important for a temporal parser since it shows the reference to the past. The corresponding tag in a task-based dependency annotation should indicate this information.

➤ **Indefinido**

The *indefinido* is used in order to refer to events in the past which ended in the past and which do not include the present time point. The event is seen as finished (Vera-Morales, 2004:340)

- (68) Hace tres días Pedro **visitó** a su abuela.
 Three days ago Pedro visited his grandmother.

➤ **Imperfecto**

The imperfect (*imperfecto*) is used for descriptions in the past tense and for repeated actions like habits (Vera-Morales, 2004:333-338). Sentence (69) exemplifies both uses. Repeated events cannot be mapped in such a simple way on a timeline. Besides the imperfect can be used for unfinished events in the past, as in (70).

(69) Cuando **era** ministro no **descolgaba** nunca el teléfono a los familiares.
When I was minister, I never answered calls from my family.

(70) Ayer a las cinco **estaba** nevando.
Yesterday at 5 pm it was snowing.

➤ **Pluscuamperfecto**

This composed tense is used to include a reference to anteriority in a past event.

(71) **Había comprado** ya las entradas, cuando me llamaste.
I had already bought the tickets, when you called me.

(72) En 2008 ya **había comprado** un apartamento.
In 2008, he already had bought an apartment.

As can be seen in (71) and (72), the event expressed by a *pluscuamperfecto* is marked with anteriority to another reference point. In (72) the event expressed by a *pluscuamperfecto* is situated in reference to a time expression (*en 2008*, in 2008), while in (71) the event in *pluscuamperfecto* is situated as prior to another event. A temporal parser identifies this tense by means of the auxiliary verb of the composed verb form. An explicit tag can therefore help the parser to find this information.

• **Future tense**

The following subsections present the future tenses used in Spanish.

➤ **Futuro perifrástico**

This future tense is composed by the use of the verb *ir* ('go'), followed by the preposition *a* ('to') and an infinitival verb form, as in (73).

(73) Mañana **va a comer** en el restaurante.

Tomorrow he will eat at the restaurant.

Again, an auxiliary verb contains important information for the temporal parser.

➤ **Futuro simple**

The future tense *future simple* refers to a time point in the future, as in (74). A temporal parser extracts this information directly from the verb.

(74) En dos meses **viajaré** a Brasil.

In two month I will travel to Brazil.

➤ **Futuro compuesto (*composed future*)**

The example (75) situates the event expressed in *future compuesto* with anteriority to a time point in the future. As seen in example (75), the auxiliary verb contains important information for the temporal parser.

(75) En dos meses **habré viajado** a Brasil.

In two month I will have travelled to Brazil.

5.1.2.2 Sequence

Having seen the different tenses used in Spanish and their effects on a task-based dependency model, the following subsections present further grammar-related phenomena, which have to be taken into account.

The sequence of verbal clauses in an action sequence gives implicitly information about the occurrences of those events as people expect the speaker to maintain the natural

order of events if not explicitly announced (see 76). A temporal parser can access this order by default.

- (76) Me senté, pedí y comí.
I sat down, ordered and ate.

5.1.2.3 Modality

Modal verbs introduce a source of uncertainty or possibility to an event, and influence in this way the factual value of the event. The event is in this way only described as merely possible or it is at least not sure that it happened. This fact has also influence on a temporal ordering of the described events as uncertain events cannot be mapped on a timeline together with events that certainly happened (Saurí, 2008).

Therefore, the effect of modal verbs has to be taken into account in a temporal analysis. The modal verbs in the examples (77) to (81) introduce a change in the certainty value of the statements. The loss of certainty is very different according to the modal verb which is used and to the context, but most cases imply such a loss. (82) exemplifies (77) without the use of a modal verb.

- (77) Diana **puede** haber ido ayer al cine.
Diana may have gone to the cinema yesterday.
- (78) El periódico, desde sus inicios, **debió** haberse vendido más.
The newspaper should have sold more since its beginnings.
- (79) **Debíamos** mantener los planos de las torres.
We should keep the plans of the towers.
- (80) **Tenía que** nombrar a los miembros del consejo.
He had to name the members of the council.
- (81) El Gobierno **tuvo que** pagar más de 544.000 euros a Pasadena Viajes.
The government had to pay more than 544,000 euros to Pasadena Viajes.

- (82) Diana fue ayer al cine.
Diana went yesterday to the cinema.

While (82) can be mapped on a timeline of happened events, the same cannot be done for the examples seen in (77) to (81) in such an easy way. A task-based dependency model should therefore mark modal verbs explicitly in order to facilitate the work of a temporal parser.

5.1.2.4 Aspect

Aspect is connected to time in different ways. In contrast to other possibilities to express temporal information (e. g. tense), aspect does not relate a situation to any other time point but gives information about the internal time structure of a situation (Comrie, 1976:5).

Semantic aspectual distinctions are made in Spanish past tense forms and give the reader additional information about the temporal order of events.

- (83) María **iba** a la iglesia y **vio** un accidente.
Maria was on her way to the church and she saw an accident.

- (84) María **fue** a la iglesia y **vio** un accidente.
Maria went to the church and saw an accident.

In (83), we do not know if María arrived at the church, we only know that she was on her way there, when she saw an accident. In (84), María actually arrived at the church and saw later the accident.

These examples show how temporal reasoning about two events can be influenced by imperfective and perfective use of Spanish verbs in past tense. While perfectivity indicates that an action was completed, imperfectivity does not.

Spanish makes also use of aspectual verbs in order to refer to the internal temporal structure of events. While the aspectual verbs in (85) and (86) denote the beginning of the event expressed by the main verb, (87) shows an example for the recent completion of it.

(85) El paciente **empieza a** tener fiebre.

The patient starts to have a temperature.

(86) **Comenzaron a** divulgar el artículo durante las elecciones.

They started to spread the article during the elections.

(87) **Acabaron de** ver una película.

They have just watched a movie.

It is therefore important for a temporal parser to take into account the information given by aspectual verbs. The planned task-based dependency model should mark this information explicitly.

5.1.2.5 Mood

The Spanish subjunctive is used in expressing one's belief or appreciation towards an event or situation (88). Nevertheless, the subjunctive can also have an effect on the concerned time of the event.

In (89), we can see how the conjunction *cuando* ('when') in combination with a present tense of the subjunctive refers to a condition which has to be fulfilled in order to have

the event of the main clause take place. One can see here the anteriority effect of the subordinate clause regarding the main clause.

If the same sentence is built with an indicative present tense form after *cuando*, the anteriority effect is prevented and the utterance expresses a general rule.

While the use of the subjunctive in (88) does not have an effect on temporal parsing, it surely does in (89) as the comparison to the same sentence with an indicative form (90) shows.

(88) Me encanta que **hayas podido venir** a mi fiesta.
I'm really happy that you made it to my party.

(89) Te doy las llaves **cuando llegues**.
I will give you the keys as soon as you arrive.

(90) Te doy las llaves **cuando llegas**.
I (always) give you the keys when you arrive.

One can see in (89) the possibility of Spanish to refer to the future by a subjunctive form and change in this way the factuality status of the expressed event since a future event cannot be seen as definitely happening. (90) in contrast is a generic statement.

In the case of the conjunction *aunque* ('even if') one can see a different usage of indicative (see 91 and 92) and subjunctive forms according to the degree of certainty of the event presented in the subordinated clause. The use of the subjunctive indicates that the event is merely a possibility (93) or did not have effect on the event of the main clause (94).

(91) Aunque me caigo de sueño, seguiré trabajando.
Even if I am really exhausted, I will continue to work.

(92) Aunque soplabá un viento frío, decidieron desayunar en la terraza.
Even if a cold wind was blowing, they decided to have breakfast on the terrace.

- (93) Suele dormirse en los conciertos aunque toquen su música preferida.
He usually falls asleep in concerts, even if they play his favourite music.
- (94) Yo no me podía creer eso aunque lo hubiera contado el propio Toribio.
I could not believe that, even if it had been told by Toribio himself.

5.2. Dependency Relations Customised to Temporal Parsing

The previous section has shown how time information is conveyed in Spanish and the following section describes now how the previous observations can be reflected in a dependency model adapted to temporal parsing. The customisation of dependency structures according to the further usage shall provide advantages in the performance of temporal parsers. The dependencies adapted for temporal parsing will provide shorter paths through the dependency tree to the relevant nodes and will make parsing easier as a first selection of important and unimportant nodes of the dependency structures is done. Furthermore, they are enhanced by rich information for this specific task. This section explains the creation of Temporal Dependencies and starts with an explanation of the chosen approach (Subsection 5.2.1). This includes a description of the way temporal information is emphasised in the model (Subsection 5.2.1.1) and a detailed overview of the implied linguistic decisions (Subsection 5.2.1.2). Afterwards, the tagset for Temporal Dependencies is discussed (Subsection 5.2.2) and information about the actual implementation of the model (algorithm, tagset and results) is given (Subsection 5.2.3).

5.2.1 Approach

As Section 5.1 has shown, there are many ways to express time information in Spanish and a temporal parser has to be sensitive to all of them. The idea for Temporal Dependencies is therefore to offer rich input data to a temporal parser in order to facilitate its work. The lexical part of the treatment will be covered by the identification of time expressions and further rules will work over other time-related lexical markers such as conjunctions and prepositions. The extraction of event information is important in temporal parsing as well, therefore the verbs, which convey semantic information, will be situated as heads of possible auxiliary or modal verbs. The tagset will be used to encode time information directly into the dependency annotation. Collapsed dependencies, as in de Marneffe and Manning (2008), enable the shortening of distances within the dependency annotation between relevant nodes and will be implemented at convenient points.

This subsection explains first how temporal information will be added to the annotation of Temporal Dependencies (Subsection 5.2.1.1) and describes then the linguistic decisions that are taken regarding head selection (Subsection 5.2.1.2).

5.2.1.1 Addition of Temporal Information

The identification of time expressions gives important information for temporal parsing and should be added in this dependency model. This can be done by means of the syntactic function tagset, which can mark explicitly relevant information. Temporal Dependencies will use a tagset that is based on the one seen for SSD, but enhanced with temporal-syntactic tags (see Subsection 5.2.2). In this way, a temporal expression can have two possible types of labels in Temporal Dependencies: a purely syntactic one and a temporal-syntactic one.

The purely syntactic one refers to the syntactic functions already seen in SSD like *nsubj*, *pobj* or *dobj*. Temporal Dependencies provide additional temporal-syntactic labels. These new labels are created basically by adding a *t* as the first character of the former

purely syntactic tags. Thus the tag *tadvcl* refers, for example, to an adverbial clause which contains time information. A full listing of temporal tags is presented in Section 5.2.2. These temporal-syntactic labels identify nodes already as time expressions, but, nevertheless, an additional tag has to be created in order to mark time expressions which do not use a temporal-syntactic label. In this case, *tmp* is added to the purely syntactic tag as additional information in order to mark the corresponding node as head of a temporal expression. The addition of *tmp* is done after the syntactic tag and with a colon in between, which indicates to the temporal parser that the tag contains further temporal information. An example can be seen in (95).

(95) **Ayer** fuimos a Barcelona.



advmod:tmp

Yesterday we went to Barcelona.

5.2.1.2 Linguistic Decisions for Head Selection

The customisation of dependencies for temporal parsing requires some changes with respect to the head selection seen in the previously prepared surface-syntax dependencies. A temporal parser needs to extract semantic information from a text. Thus, the verbs which carry semantic information should have a good position in the dependency structure. Furthermore, the previous section has shown that a temporal parser needs to make use of important information in auxiliary, modal and aspectual verbs. This information should be marked explicitly by the tagset. The general idea for the task-based model is also to shorten distances in the dependency structure to the important nodes. Therefore, Temporal Dependencies will make use of collapsed dependency structures, which disconnect nodes of little relevance for the task in benefit of shorter distances between important nodes.

The following subsections describe these changes structure by structure and take a look at the reasons why these changes are preferred to the surface-syntax ones for the purpose of temporal parsing.

- **Composed Tenses**

In composed verb tenses it is preferable to set the verbal form carrying the semantic content as the head of the construction for temporal parsing since the verb which contains the semantically important information is the one denoting the event to be identified by the parser. The examples in (96) and (97) show cases in which the treatment is different compared to SSD.

(96) El sospechoso **se ha marchado** a las 10 de la noche.



The suspect has left at 10pm.

(97) El teléfono **fue comprado** el 5 de enero de 2004.



The telephone was bought on 5th January 2004.

In (97), for example, the auxiliary verb form *fue* ('was') is the head of the verb structure in SSD, while the verb form *comprado* ('bought') is the head in Temporal Dependencies, as it carries the semantic information.

Temporal information is normally linked to an event expression which in this way can be anchored on the timeline. The semantic head refers to the described event and its node is preferably set higher in the dependency tree. This makes temporal parsing more effective and reduces parsing costs and the possibility to miss events in the parsing process. Section 5.1 has shown that a temporal parser also needs to know about auxiliary verbs. Therefore the auxiliary verbs will be directly connected to the head verb and be marked as auxiliary by the tag *aux*. In this way, a temporal parser can easily extract the relevant information.

- **Modal and Aspectual Verbs**

The treatment of modal verbs is similar to the one described for composed tenses as the verbal form which contains the semantically relevant information is the form preferred as head of the verb construction.

The modal or aspectual verb contains the syntactically important information since it is the conjugated verb of the structure but not the semantic relevant information about the described event. Therefore the one containing the latter is preferred as head of the structure.

(98) **Acabo de enterarme** de una triste noticia.



I just got to know some sad news.

(99) **Empieza a dictar** la incapacidad jurídica de algunas.



He begins to dictate the legal incapacity for some.

(100) Esa generosidad quijotesca la **debió heredar** de su abuelo.



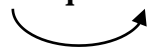
He had to inherit this quixotic generosity from his grandfather.

In (100), for example, the modal verb form *debió* ('had to') would be head of the verb structure in SSD, but the infinitive *heredar* ('inherit') in Temporal Dependencies as the latter carries the semantic information.

Modal and aspectual constructions which include a preposition or conjunction can drop those as they are not of importance for temporal parsing. This treatment leads to shorter paths and does not lose important information as the relation between infinitive and modal/aspectual verb can contain the dropped preposition name (102). This is a technique seen already in other works such as Stanford Collapsed Dependencies (de Marneffe and Manning, 2008). Section 5.1 has shown that modal and aspectual


information is also important for a temporal parser. Therefore, the modal and aspectual verbs will be directly dependent on the verb which carries the semantic information and marked as tags *modal* and *asp* (aspectual), respectively. In this way, a temporal parser can easily extract the relevant information. (101) and (102) exemplify also the tags that these constructions take in the Temporal Dependencies model.

(101) **Tiene que afrontar** los costes del retiro.


modal|que

He has to defy the pull-off's costs.

(102) **Acabo de enterarme** de una triste noticia.


asp|de

I just got to know some sad news.

- **Subordinated Clauses**

This subsection gathers several types of subordinated clauses.

- **Relative Clauses**

The treatment of relative clauses in the surface-syntax approach is kept since the head of the subordinate clause was already its main verb and not the relative pronoun (103).

(103) Se hablaba de un hombre **que era** capaz de decapitar una rata.



There were rumors about a man who was able to decapitate a rat.

This treatment is preferable as the relative pronoun does not contain important information for temporal parsing and the selection of the verb as head makes the path shorter to the described event and to possibly connected temporal information.

➤ **Completive Clauses**

The treatment of completive clauses will be similar to the one chosen for relative clauses. The main verb of the completive clause is the head and the conjunction its dependent.

(104) Cree **que** el balonmano **necesita** equipos como este.



He thinks that handball needs teams like this one.

(105) Ha dicho **que** vendrá.



He said that he will come.

As both (104) and (105) show, the verbs which carry semantic information are directly connected in Temporal Dependencies and facilitate temporal parsing purposes in this way.

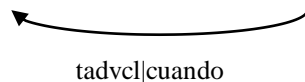
➤ Adverbial Clauses

Adverbial clauses will be divided in this section into temporal clauses and non-temporal clauses, as the former are especially important for temporal parsing.

▪ Temporal Clauses

The conjunction is disconnected from the dependency tree, and main and subordinate clauses are linked through their main verbs. SSD have the conjunction as head, but in a temporal analysis it is preferable to have the described event as the head. The conjunction can then be added to the syntactic function tag of the relation between the two verbs in order to not to lose any important information. Additionally, the relation will be named *tadvcl* instead of the previous *advcl* so that the parser detects easily that it contains relevant temporal information. Example (106) shows this treatment for a temporal clause.

(106) La mujer estaba en Roma **cuando** nació su hijo.



The woman was in Rome when her son was born.

In Spanish, temporal subordinated clauses are marked lexically by their conjunction. There are several temporal conjunctions which explicitly introduce such a temporal clause. The indicated time point in the temporal clause can be divided into anteriority, simultaneity and posteriority. The following figure shows examples for all three of them.

<u>anteriority</u>	<u>simultaneity</u>	<u>posteriority</u>
después de (after)	cuando + gerund (when)	después (after)
cuando + ant. marker (when)	mientras (que) (while)	hasta (que) (until)
desde (que) (since)	durante (while)	antes de (que) (before)
antes (before)		
en cuanto (when)		
tan pronto como (as soon as)		
apenas ⁶ (as soon as)		

Table 19: Markers for temporal clauses

Note that the mentioned temporal conjunctions have to be connected to a verb in order to be considered a temporal clause. A prepositional phrase like *desde la oficina* (‘from the office’) is not included in this consideration.

▪ **Non-Temporal Clauses**

Non-temporal clauses are treated as the completive clauses: the conjunction is not the head anymore but dependent to the verb of the subordinate clause. The conjunction is not kept in the relation of the two verbs. In this way, the verbs are heads of the two clauses and directly connected. The relation between the two verbs will be named *advcl* as before and shows the parser thereby that it is not a temporal clause. The example in (107) shows how the conjunction is now dependent of the verb in the subordinate clause. In this way verbs carrying semantic information connect directly (in this case: *va* and *evolucionado*) and facilitate information extraction in temporal parsing tasks.

⁶ in Latin America

(107) Va en aumento **porque** los coleccionistas han evolucionado.



conj

It is increasing because the collectors have changed.

• Coordinations

The sequence of events is important in a temporal analysis since humans normally describe events in an ordered way and not in a random sequence. In (108), the reader extracts information with an implicit temporal order, therefore the order in which conjuncts are expressed within a coordination structure can also be of importance for a temporal ordering of those events.

(108) **Cenó, se lavó** los dientes y **fue** a la cama.

He ate dinner, brushed his teeth and went to bed.

The first column in the surface-syntax annotation states already the position within the sentence (see Figure 21). So this treatment can be kept in the temporal annotation.


1	Casi	casi	rg	rg	_	2	advmod
2	todos	todo	di0mp0	di0mp0	_	4	det
3	los	el	da0mp0	da0mp0	_	4	det
4	expertos	experto	ncmp000	ncmp000	_	20	nsubj
5	que	que	pr0cn000	pr0cn000	_	6	nsubj
6	han	haber	vaip3p0	vaip3p0	_	4	rmod
7	analizado	analizar	vmp00sm	vmp00sm	_	6	vobj
8	sistemáticamente	sistemáticamente	rg	rg	_	7	advmod
9	el	el	da0ms0	da0ms0	_	10	det
10	contenido	contenido	ncms000	ncms000	_	7	dobj

Figure 21: SSD dependencies format example

The general treatment of the syntactic analysis of coordinations stays the same as in SSD with the coordinating conjunction as head of the coordination structure.

- **Comparatives**

In comparative structures, the conjunction of the embedded phrase (e. g. *que*, ‘that’) is disconnected and the embedded item becomes dependent of the first item (e. g. *barato*, ‘cheap’). The function name *crobj* is kept as in the surface-syntax version and set as relation between the embedded items.

(109) Es **más** barato **que** bueno.

crobj|que

It is cheaper than it is good.

The advantage of this treatment for the purpose of temporal parsing is that the path gets shorter by the eliminated conjunction. Furthermore we keep in this way the treatment of not having a conjunction as head of a relation.

- **Prepositional Phrases**

Two different types of prepositional phrases are considered. Those which contain events and those which do not. Note that this approach only considers verbal events and temporal expressions. It does not identify nominal events.

➤ **Containing event-denoting expressions**

The preposition will be disconnected in the dependency tree and the syntactic function tag changes to *tprpv* plus the name of the preposition.

(110) Se conocen **desde que fueron** a Sevilla.



tprpv|desde que

They know each other since they went to Seville.

(111) No le ha visto **desde el mes** pasado.



tprpv|desde

She has not seen him since last month.

In (110), a verbal event and its treatment in Temporal Dependencies is shown with *fueron* ('went') and in (111) an example for a time expression.

Since the detection of events is limited in this work to the identification of verbs and time expressions, nominal events like *caída* ('fall') in (112) will not be considered.

(112) La humanidad no había vivido nada semejante **desde la caída** del Muro.

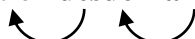


Humanity had not lived anything similar since the fall of the Wall.

➤ **Not containing event-denoting expressions**

In this case the preposition will not be dropped and the surface-syntax treatment is kept.

(113) Muchos de ellos viajaron **desde Barcelona**.



prepv pobj

A lot of them travelled from Barcelona.

The example in (113) shows the treatment of this type of prepositional phrases for both Surface Syntax and Temporal Dependencies.

➤ **Combination of prepositions**

The treatment seen for prepositional phrases can also include preposition structures which span over several tokens. All words will then be disconnected and added to the function tag, as seen in (114). This treatment brings the events closer together and facilitates an event-oriented parsing, as needed in temporal parsing.

(114) Pablo habló con su madre **antes de** salir.



tprep|antes de

Pablo spoke to his mother before he left.

➤ **Infinitival complements**

Infinitival complements show direct connections between infinitival verb and the upper head (see 115-117). All connectors in between are dependents of the infinitival verb. This treatment facilitates the event extraction of a temporal parser.

(115) Pablo **piensa en volver**.



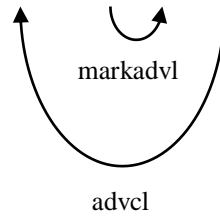
Pablo thinks about his return.

(116) Pablo **piensa en que volverá**.



Pablo thinks of his return.

(117) Como no encontraba **plaza para estudiar...**



As he did not find a place to study...

5.2.2 Temporal Dependencies Tagset

The Temporal Dependencies tagset is an enriched version of the SSD tagset. It contains some more tags than the former, which is mainly due to the fact that rich information for temporal parsing has been added to the syntactic function tags. Some existing tags can be enriched by a *t* in first position to introduce a temporal relation to the tag. Thus a temporal expression which is used as a direct object in a sentence would therefore convert from a *dobj* tag in SSD dependencies into a *tdobj* tag and express this way that it is used as a *temporal direct object*. The adding of a temporal *t* to tags is not constrained but many tags are unlikely to refer to temporal expressions. There is no *tdet*, for example, as Spanish does not include anything such as a *temporal determiner*.

Table 20 shows the list of all tags in the Temporal Dependencies Tagset; it includes those *temporal tags* which correspond to linguistic structures which can contain temporal expressions in Spanish.

SSD Tag	Temporal Dependencies Tag	Full name
root	root	root
dep	dep	dependent
	tdep	temporal dependent
arg	arg	argument
	aux	auxiliar
	asp	aspectual verb
	modal	modal verb
comp	comp	complement
attr	attr	attributive

cpred	cpred	predicative complement
obj	obj	object
cobj	cobj	complementizer object
dobj	dobj	direct object
	tdobj	temporal direct object
iobj	iobj	indirect object
	tiobj	temporal indirect object
oobj	oobj	oblique object
	toobj	temporal oblique object
pobj	pobj	object of a preposition
vobj	vobj	object of verb
crobj	crobj	object of comparative
	mark	marker
	markobl	oblique marker
	markadvl	adverbial marker
	compl	complementizer
subj	subj	subject
nsubj	nsubj	nominal subject
	tnsubj	temporal nominal subject
csubj	csubj	clausal subject
coord	coord	coordination
	tcoord	temporal coordination
conj	conj	conjunct
agent	agent	agent
reflec	reflec	reflexive (“se”)
te	te	textual element
mod	mod	modifier
abbrev	abbrev	abbreviation modifier
amod	amod	adjectival modifier
appos	appos	appositional modifier
advcl	advcl	adverbial clause modifier
	tadvcl	temporal adverbial clause
det	det	determiner
infmod	infmod	infinitival modifier
partmod	partmod	participial modifier
advmod	advmod	adverbial modifier
neg	neg	negation modifier
rmod	rmod	relative clause modifier
nn	nn	noun compound modifier
tmod	tmod	temporal modifier
num	num	numeric modifier
	tnum	temporal numeric modifier
prep	prep	prepositional modifier
prepv	prepv	prep. mod. of a verb
prepn	prepn	prep. mod. of a noun
prepa	prepa	prep. mod. of an adjective
	tprep	temporal prep. modifier

	tprepv	temporal prep. mod. of a verb
	tprepn	temporal prep. mod. of a noun
	tprepa	temporal prep. mod. of an adjective
poss	poss	possession modifier
punct	punct	punctuation
	–	disconnected from the dependency structure

Table 20: Temporal Dependencies tagset

The tag *tadvcl* shows that a *temporal adverbial clause* has been identified. Note that this does not necessarily refer to an adverbial clause which contains a temporal expression. A conjunction like *cuando* (‘when’) can introduce such an adverbial clause without the inclusion of a temporal expression.

Additionally, seven tags are added in order to increase the informative value of the annotation. Three of the tags are introduced due to the focus on semantic verb information. These three new tags are *aux*, *asp* and *modal* and give detailed information about the type of auxiliary verb that is connected to the main verb which contains the semantic information. The other four added tags are conditioned by the treatment of adverbial clauses and oblique objects as in both cases the verb containing the semantic information is now head of the corresponding structure. The tags are the general tag *mark*, and the three finer-grained tags *compl*, *markobl* and *markadvl*.

While the total number of tags climbs up to 62, it has to be taken into account that several tags are not actually used in this annotation. Some tags serve merely as general tag in case that no other finer-grained tag can be chosen (such as *mark*), other tags have not been used by the algorithm (such as *infmod*)

The following list gathers all not used tags by the algorithm:

arg	subj
comp	nn
obj	infmod

Table 21: Tags without usage in Temporal Dependencies annotation

It should be taken into account that the creation of other, erroneous, tags is possible due to the attachment of a *t* by the algorithm in case of temporal expressions. A node which is wrongly identified as a temporal expression will attach a *t* to its function even if it should not be a temporal tag. The error analysis in the results section (see Subsection 5.2.5) gives further details.

5.2.3 Implementation

This section describes first the algorithm that is used for the conversion from SSD into Temporal Dependencies, and explains afterwards how the time expressions are identified and what kind of rules have been created to implement automatically the chosen linguistic criteria into the annotation.

5.2.3.1 Algorithm

The algorithm (see Figure 22) starts reading SSD dependencies sentence per sentence (line 2). Each sentence is passed through ENTimex-Recognizer, a module that identifies time expressions (line 3) and which is explained in the next subsection. Information about the identified time expressions and the sentence itself is then used to adapt the sentence to the chosen linguistic criteria. This is done by help of rules, which are mainly based on part-of-speech and head information, and by use of the time expression identification (line 4). Finally, Temporal Dependencies are annotated (line 5).

```
1 function TIME_DEPENDENCY_ANNOTATION(ssd_corpus):  
2 for ssd_sentence in ssd_corpus:  
3     timexes=Recognizer(ssd_sentence)  
4     time_sentence=general_adaptations(ssd_sentence,timexes)  
5     write_time_dependency_annotation(time_sentence)
```

Figure 22: Temporal Dependency Annotation algorithm

5.2.3.2 ENTimex-Recognizer

ENTimex is a system for the detection and normalisation of time expressions which was created as my Master's thesis project (Kolz, 2012). The system contains two components, on the one hand Recognizer, a time expression identifier, and on the other hand Normalizer, a time expression normaliser. The current automatic annotation of Temporal Dependencies makes use of Recognizer, since the nodes which are identified as time expressions are treated in specific ways. Normalizer has not been used, since its task is not necessary for the model adaptation.

The time expression recognition task concerns the exact detection of a temporal expression in natural language text. Time expressions can be seen as those constructions referring to points or intervals on the timeline (Saurí, 2010:3).

For this undertaking, the starting and end points of the time expression have to be found. Additionally, the type of temporal expression can be determined. While Recognizer does both these individual identification tasks, information about the type of time expression is not necessary for the creation of Temporal Dependencies.

Time expressions can be expressed by different parts of speech such as temporal adverbials (e. g.: *semanalmente*, 'weekly') or nouns referring to weekdays, holidays (e. g.: *Navidad*, 'Christmas') etc. In (118), *el lunes* ('the Monday') would be marked as temporal expression.

(118) El Barcelona podría haber jugado **el lunes**.

Barcelona could have played on Monday.

Recognizer identifies in each sentence of the input data all time expressions at their full span within the text and returns a list of those, so that the automatic Temporal Dependencies annotation takes that list as input.

Note that Recognizer showed very good results when it was evaluated, but that it does not necessarily identify all time expressions correctly in the input data. The evaluation

results from the Master’s thesis are shown in Figure 23; the precision was really high with a 0.92 and also a recall of 0.83 seems to be a good result. Nevertheless, it also shows that not all time expressions will be found and this will also influence the automatic annotation of Temporal Dependencies since not identified time expressions cannot be treated as planned.

Results for ENTimex-Recognizer:	
precision	0.92
recall	0.83
F-Measure	0.87

Figure 23: Recognizer performance

5.2.3.3 Rules

A set of rules has been created to conduct the changes described in Subsection 5.2 according to the chosen linguistic criteria. The rules are based mainly on head, lemma, part of speech and time expression information. Additionally, they have access to the syntactic function in SSD.

So, for example, if a node is a verb with *advcl* (adverbial clause) as syntactic function in SSD and the head’s lemma is *cuando* (‘when’), then the verbal node will now be the head of the conjunction *cuando* and the syntactic function of the verbal node is *tadvcl/cuando* from now on, while the node corresponding to the conjunction is disconnected from the dependency structure.

Time expressions are marked by the addition of *:tmp* to the syntactic function tag, if they are not already marked by a *temporal tag* (such as *tmod*, *tadvcl*, etc.).

In total 36 rules have been implemented which handle the adaptation of SSD to Temporal Dependencies, but it has to be taken into account that some rules handle a variety of different lexical markers which enter into the same pattern. One rule handles, for example, six adverbial modifiers with time related meaning.

The rule set adjusts head and syntactic functions of all nodes according to the established linguistic criteria for Temporal Dependencies and enhances rich information for temporal parsing.

5.2.4 Customisation Results

This section explains first the creation of the evaluation corpus, presents then the obtained results and, finally, discusses some observed errors and possible solutions for those.

5.2.4.1 Evaluation Corpus

The algorithm for the customisation to Temporal Dependencies takes as input AnCora Surface Syntax Dependencies and therefore the system has been evaluated by help of a manually annotated evaluation corpus based on the input corpus.

The evaluation corpus consists of 21 randomly gathered files from AnCora Surface Syntax Dependencies, which have been manually adapted to the linguistic criteria chosen for Temporal Dependencies. This was done by two professional linguists, which annotated each half of the evaluation corpus. Additionally, complex structures were discussed between both of them and a common agreement reached.

For this evaluation, only nodes were taken into account which implied a modification in comparison to the input corpus which is AnCora Surface Syntax Dependencies. This implied a total of 870 tokens which had to be modified compared to the base corpus. A token counts here as modified if head, function or both of them have been changed compared to the base model.

5.2.4.2 Result Data

The results of the temporal modal adaptation are presented in Table 22. The general adaptation has been measured in terms of Labelled Attachment Score (LAS), which implies the correct value for both head and syntactic function label, Unlabelled Attachment Score (UAS), which measures only the correctness of heads, and finally with Label Accuracy (LA), which on the other hand measures correct syntactic label values.

	UAS	LA	LAS
Evaluation corpus	0.85	0.83	0.80

Table 22: General adaptation results for Temporal Dependencies.

The temporal modal annotation shows a UAS of 0.85 considering the head identification task for the dependency nodes and a LA of 0.83 in the setting of the correct syntactic function tag between a head-dependent pair.

If only those dependencies and syntactic functions are considered which had to be adapted and which were correctly set in the input data, the precision of the temporal model adaptation increases to 0.91 for both head identification (UAS) and setting of syntactic function task (LA).

The correct adaptation implied on the one hand the identification of time expressions and time-related discourse markers, and, on the other hand, the collapsing of dependency structures and thus the disconnection from the tree of unimportant nodes considering the task of temporal parsing.

In the evaluation corpus, there were 93 nodes to be identified as expressing temporal meaning, and 84 of those were correctly identified. 9 nodes were not identified and 6 nodes were falsely identified by the program. The program used Recognizer for the temporal expression identification and further rules for the identification of further temporal markers which according to TimeML are not considered temporal expressions.

The dependency collapsing process showed also good results as 81 nodes were correctly disconnected, while only 5 nodes should have been disconnected but were not identified as such and 5 nodes which were disconnected and should not have been so.

The automatic annotation made use of 62 tags, even if some tags of the tagset were not used, while other temporal tags were erroneously added. The not used tags include *tiobj*, which theoretically could have been used by the algorithm, but which did not show any instance in the automatic annotation of AnCora Temporal Dependencies and the underspecified tags, which were commented in 5.2.3. On the other hand, 7 tags were created erroneously and are further discussed in the next subsection.

5.2.4.3 Error Analysis

This subsection addresses several points to further improve the performance of the automatic adaptation process. The first division of the found errors can be made into source-related and adaptation-related errors. It has to be taken into account that errors in the input data influence highly the correctness of the adaptation's output data.

The general adaptation to Temporal Dependencies has shown problems when handling complex coordination structures (f. ex. *Dado que... y que...*, 'Given that... and that...'). This is on the one hand due to problems with these structures already in the input data, which obviously brings up consecutive errors, and, on the other hand, on the rules which handle coordination structures and which were not specific enough to handle more complex coordinations.

Input errors have also a strong effect on the Temporal Dependencies annotation since the treatment of conjunctions was changed regarding the SSD model and conjunctions like *cuando* ('when') directly affected a time-related treatment. In Chapter 4.3, flat structures have been commented, which mostly occur in connection with this kind of conjunctions and problems with them lead to consecutive errors within Temporal Dependencies.

Another type of errors is due to the identification of time expressions. While the system Recognizer has shown really good results, the automatic identification of time expressions is still a source of errors, which is reflected in the final AnCora Temporal Dependencies annotation. In the evaluation corpus there was, for example, a wrongly identified *diario* (...*publica el diario*, ‘... publishes the newspaper’), a word which according to the context can have a time related meaning (‘daily’) but also mean *newspaper*. It can also happen that time expressions are correctly identified by Recognizer but not with its complete span. An example is *a mediados de año* (‘at midyear’), in which the system only recognized *año* (‘year’) as time expression. This false span brought up an error in the treatment of Temporal Dependencies.

The tagset has shown some problems regarding the creation of *temporal tags*. As it is theoretically possible for the algorithm to create *temporal tags* for all available syntactic tags, errors in the syntactic analysis can easily lead to wrong tags and to the creation of wrong *temporal tags*. The proper name *Julio Medem*, for example, created a *tappos* tag in the corpus due to an error in the identification of temporal expressions. The first name *Julio* was identified as the name of the month by the system and brought up the *tappos* tag in this way. The following list (Table 23) presents the erroneously created tags in the automatically annotated corpus.

tattr	(3)	trcmo	(1)	tcobj	(9)
tpobj	(17)	tappos	(1)		
tpunct	(3)	tconj	(3)		

Table 23: Erroneously created tags (frequency in brackets)

As Table 23 shows, the frequency of those erroneous tags was nevertheless very low and has not had a strong influence on the overall quality of the automatic annotation. The tag *tpobj* has the highest frequency in this list and is due to problems with coordinations of time expressions which connect to prepositions. In *La operación tendrá lugar entre el 5 y 17 de enero* (‘The operation will take place between the 5th and 17th of January’), the coordination *y* (‘and’) was marked erroneously with *tpobj* instead of *tprepv* and the preposition *entre* (‘between’) not disconnected from the dependency structure.

The different types of errors have shown that it is first of all important for the adaptation process to receive correct input data since consecutive errors will come up otherwise. The adaptation algorithm cannot correct those. Nevertheless, several aspects have also been found which can be improved in the algorithm, such as a higher control of the creation of *temporal tags* and the creation of rules for more complex coordination structures. The error analysis can lead to an improved version of the adaptation program in the future.

5.2.4.4 Conclusion

The creation of a dependency model adapted to temporal parsing has been completed successfully. Temporal Dependencies includes a task-adapted head selection, the use of collapsed dependencies for shorter distances between event and time information, an enhanced tagset with specific time-related tags, and the labelling of detected time expressions.

The model has already seen a practical use by an automatic adaptation of the base model SSD annotation into AnCora Temporal Dependencies. The created corpus represents now an optimised and enriched input for NLP applications, which make use of time information.

Regarding the quality of the annotation, one should be aware that already AnCora Surface Syntax Dependencies has been an automatic conversion from constituents and served as input data for the Temporal Dependencies adaptation. Errors in the input data will most likely lead to errors in the output. This source of errors exists and should be taken into account. As seen throughout this chapter, temporal parsing is a complex task. Spanish has many ways to express time information and the automatic annotation has to fulfil many different requirements. Nevertheless, AnCora Temporal Dependencies has shown to be a dependency annotation of high quality and offers an important resource for future linguistic projects.

6. Discourse Dependencies

In Chapter 5, the first task-based dependency model has been constructed successfully. This chapter is now dedicated to the second dependency model. It is called Discourse Dependencies and adapted to the NLP task of discourse parsing.

Writing a discourse is a complex task even for humans as one can see in the fact that politicians normally have somebody contracted as help for their discourses. It is a useful resource to encode a lot of information “between the lines” which refers normally to meaning implied by discourse but not necessarily expressed by lexical means. The automatic creation and understanding of discourse is nowadays still a topic of research in NLP (da Cunha, 2013; Wei Fen and Hirst, 2014; Iruskieta and Zapirain, 2015). It is lately getting more importance due to the increasing use of human-machine communication, like virtual assistants on mobile phones (e.g. Apple’s Siri) or web pages (e.g. IKEA).

Discourse Dependencies shall be designed as a dependency model, which facilitates the work of discourse parsers by offering them dependency annotations with linguistic criteria adapted to the needs of this task and enhanced by relevant information at discourse level. The optimisation process will therefore be conducted at two levels. First, the head selection will be established so that important nodes for a discourse parser are reachable by short paths and, secondly, the tagset will be used to introduce discourse relevant information directly to those important nodes.

As stated in Chapter 2, the state-of-the-art regarding discourse parsing and discourse representation gathers a considerable quantity of linguistic research. Discourse Relations (Hobbs, 1985; Wolf and Gibson, 2006) is the representation approach that will be used to enhance discourse dependencies. This decision is mainly due to the fact that the chosen framework shall be based on the linguistic structure of texts, fitting easily in a dependency representation approach; other, more logically oriented, approaches would only be convenient if a more semantically oriented representation was aimed at in this project. RST (Mann and Thompson, 1986) has seen a wide use in discourse parsers, but it uses a set of 23 relations, which is rather fine-grained and cannot be managed with the sole syntactic information available in this thesis. Wolf and

Gibson (2006) gather a set of 11 relations, which are informative enough for discourse parsing purposes. Actually, even those 11 relations are too fine-grained for the present project since the aim is not to fulfil the work of a discourse parser, but to enhance dependencies with discourse information in order to facilitate discourse parsing afterwards. Therefore, this work takes Wolfs and Gibson's relations set as starting point and adapts it according to the needs of the proposed dependency model.

The structure of this chapter is similar to the one seen for the temporal model in the previous chapter. It introduces the ways in which discourse is constructed in Spanish (Section 6.1) and the theoretical approach taken to the representation of discourse relations. Then it explains with details the creation of Discourse Dependencies (Section 6.2). The explanation includes the linguistic criteria on which the head selection is based, the chosen set of discourse relations and the actual implementation of the algorithm, together with the obtained results. The terms *nucleus* and *satellite* are used throughout this chapter to denote in a relation between two segments, the principal segment (*nucleus*) and the segment which depends on it in this relation (*satellite*).

6.1 Linguistic Approach to Discourse Parsing

This section gives some basic ideas about the construction of discourse in Spanish (Subsection 6.1.1) and explains the theoretical approach taken in this dissertation for the identification of discourse relations (Subsection 6.1.2).

6.1.1 Discourse Construction in Spanish

Spanish has many ways to express discourse-related meaning. Some means are easy to detect and other means are far more complicated to identify. In (119), the conjunction *porque* ('because') is an explicit lexical marker which is used to structure a discourse.

(119) La bebe está llorando **porque** tiene hambre.

The baby is crying because she is hungry.

In this case, the conjunction tells the reader that the upcoming subordinate clause states the cause of the event in the main clause to happen. The conjunction can therefore be identified as an explicit lexical discourse marker.

If one considers the same example just with a comma and without the conjunction, the meaning of the sentence still stays the same (120).

(120) La bebe está llorando, tiene hambre.

The baby is crying, she is hungry.

Nevertheless, the structure of the discourse is implicit in this case. This makes it actually more difficult for the reader to connect the events in the text with each other, even if it is still a common writing style.

The same content found in both examples in one single sentence could also be split over two sentences (121).

(121) a. La bebe está llorando.

The baby is crying.

b. Tiene hambre.

She is hungry.

In this case, the event of b has to be related to the event of sentence a. This means a discourse analysis has to work across sentence boundaries (Wei Feng and Hirst, 2012) and cannot always be based on explicit lexical markers (Pitler, E. et al., 2008).

Another difficulty comes up in discourse analysis due to ambiguity of lexical markers. There are two types of ambiguity (Pitler and Nenkova, 2009). On the one hand, there are lexical elements which can be discourse markers or have a non-discourse structuring meaning (122), and, on the other hand, there are lexical markers which can introduce

different kinds of discourse relations (123). The disambiguation of lexical discourse markers is not a trivial task (da Cunha, 2013).

(122) a. Te doy las llaves **cuando** vengas.

I will give you the keys when you arrive.

b. El chico aplaude de vez en **cuando**.

The boy applauds from time to time.

(123) a. Te doy las llaves **cuando** vengas.

I will give you the keys when you arrive.

b. **Cuando** llegaba a la puerta, escuchó un ruido extraño.

When he arrived at the door, he heard a strange noise

In sentence (122a) *cuando* ('when') is used to connect the main and subordinate clauses and shows that the subordinate clause contains a condition on the event in the main clause. In sentence (122b) *cuando* is used inside the adverbial expression *de vez en cuando* ('from time to time') and does neither connect clauses nor introduce any discourse meaning.

In (123), both usages of *cuando* ('when') do connect clauses and both times a discourse relation between those clauses is introduced by means of the conjunction. Nevertheless, those relations are not of the same type. In sentence (123a) one sees a condition on the main clause to happen, in sentence (123b) the reader gets to know the sequence of the events.

The parsing of discourse is therefore a difficult undertaking. An automatic parsing approach needs to take into account a considerable amount of linguistic resources.

6.1.2 Discourse Dependencies Relations

This section presents the approach taken for discourse relation identification between segments in the text. Cross references to state-of-the-art discourse relation sets are given to facilitate the understanding of the present approach.

6.1.2.1 Introduction

The discourse relations used in this annotation are closely related to common works in the field (Hobbs, 1985; Wolf and Gibson, 2006) but show a more abstract approach to discourse relations. This is done in this way since the discourse dependency model presented here is not intended to do the full work of a discourse parser, which would be necessary to identify such fine-grained discourse relations at a decent quality. Discourse Dependencies rather present a rich input for discourse parsers, which can subsequently convert the “abstract” discourse relations into finer-grained discourse relations if needed. Discourse parsers are expected to improve their results if they take these adapted dependencies as input, compared to using a non-optimised input.

The following subsection takes a look at the used discourse relation tags and explains their meaning. In the proposed discourse dependency model the syntactic function tags are informed with the discourse relation tags: they are attached to the head of the segment which evokes them. Furthermore, the explicit lexical discourse marker, which evokes the relation, is disconnected from the dependency structure and added to the syntactic function tag of its head by means of a | plus the word form. See an example of this annotation form in (124).

(124) Word form Syntactic function tag
compraron root:opposition|Aunque

It has to be mentioned that the dependency model only works with explicit discourse markers and at sentence level. It does not identify which sentences interact with each other. The work at text level is very important for a discourse analysis (da Cunha, 2012, Wei Feng and Hirst, 2012), but is left for a discourse parser, which can benefit from the identified relations at sentence level in order to establish a full text discourse analysis.

6.1.2.2 Relations Set

Discourse Dependencies Relations (DDR) are based on the works done by Hobbs (1985) and Wolf and Gibson (2006). Table 24 lists the DDR discourse relations indicating how they map into Hobbs' and Wolf and Gibson's sets of relations to facilitate the understanding. As can be seen in the table, not all discourse relations used by Wolf and Gibson were already present in Hobbs' work. Both *attribution* and *same* were not considered by Hobbs. The table marks those that have no match with a hyphen. It should be pointed out that *condition* was treated by Hobbs but not distinguished from the *cause-explanation* cases. Both Hobbs and Wolf and Gibson are given as comparison since the former built up the base for discourse relations with his work (1985) and the latter created an updated look on the topic, taking into account past works on this research field (such as Mann and Thompson, 1986), and seeing discourse relations from the more practical point of view of discourse parsing. Nevertheless, the resulting discourse-related tagset is not only compatible to Wolf and Gibson. It can be also used for other frameworks, such as RST.

Discourse Dependencies Wolf and Gibson Hobbs Relations

Enumeration	Temporal Sequence, Similarity	Occasion, Parallel
Inclusion	Elaboration, Example, Generalisation	Evaluation, Background, Exemplification
Opposition	Violated expectation, Contrast	Violated expectation, Contrast
Causality	Condition, Cause-effect	- , Cause / Explanation
Attribution	Attribution	-
Same	Same	-

Table 24: Discourse relations overview

DDR will use a more abstract and more reduced set of relations than the former works. The idea is to gather semantically related relations into a superset. This superset can then be directly used for discourse parsing purposes or later be converted by a discourse parser into finer-grained discourse relations, when all contextual, discourse related, information is available. In this way, DDR presents a compact and informative relations set, which can offer a high-quality automatic dependency annotation without making use of a full discourse parser. As shown in Table 24, DDR works with a set of six relations, while Wolf and Gibson make use of eleven. An example of the semantic closeness of relations in Wolf and Gibson is the pair of relations *violated expectation* and *contrast*. Both relations relate a specific content with information that is somehow opposed to what the reader would expect from the context. The next subsections describe with details and examples the set of relations that is proposed in this dissertation.

- **Enumeration**

This relation holds between two events, which are presented together within the same sentence or in different sentences, and are temporally ordered. Additionally, also segments are included which are presented in a similar way within the same sentence or across sentences and which may have an event in common. In the former case, the temporal order will not be determined but the tag *enumeration* shows only that there exists a temporal order between two or more events.

In Spanish the temporal order can be established explicitly by the use of temporal expressions, as in (125), but also by merely descriptive sentences (126) since the reader of the text will imply that the discourse is coherent, and therefore temporally ordered if necessary for the comprehension.

The explicit ordering can be performed by help of adverbs (127) like *primero* ('first'), *después* ('afterwards'), *además* ('additionally') or *más tarde* ('later'), and conjunctions

(128) like *después de que* ('after'), *antes de que* ('before') and *mientras* ('while'). The implicit order comes normally together with the use of commas or the coordination *y* which both help to order a discourse. In the following examples brackets [] are used in order to show segment limits and facilitate the reading.

(125) [Miguel Delibes ha fallecido **hoy** en Valladolid] [**y mañana** será enterrado.]

Miguel Delibes has died today in Valladolid and tomorrow he will be buried.

(126) [Cenó,] [se lavó los dientes] [**y** se fue a la cama.]

He ate dinner, brushed his teeth and went to bed.

(127) [**Primero** tengo que comprar la comida,][**después** puedo cocinar.]

First, I have to buy the food, then I can cook.

(128) [El Gobierno indonesio ha lanzado una alerta de tsunami] [**después de que** se registrara el temblor.]

The Indonesian government has launched a tsunami warning after the tremor was registered.

Both explicit and implicit cases of temporal ordering will be annotated with the *enumeration* relation. In case of explicit markers they will be disconnected from the tree and added to the syntactic function of the head. It is worth commenting that sentences like (126) do not have an explicit marker for temporal ordering but do have an explicit discourse marker for enumeration (in this case a coordination).

A temporal sequence is just one case of the *enumeration* relation. The latter does not always imply a temporal order of the introduced events. One can find *enumerations* as a kind of listing of events without temporal order which only have an event in common.

Examples (129) and (130) show how adverbs like *además* ('additionally') can be used to mark explicitly an *enumeration* relation. As seen in (129), the coordination *y* ('and') already introduces this relation and the adverb *además* additionally emphasises this relation. In (130), on the other hand, one can see an example for the *enumeration* relation across sentence boundaries, where *además* is the explicit marker for this relation.

(129) Esperaban echar un pasodoble **y** les cayó **además** un mitin político.
 They wanted to dance the pasodoble and, additionally, a political meeting came up.

(130) a. El agente ha sido trasladado al hospital comarcal, donde se encuentra en estado crítico.

The agent has been transferred to the regional hospital, where he is in critical condition.

b. **Además**, hay otra persona herida en el mismo suceso y que está ingresada en el hospital.

Besides, there is another person who was injured in the same incident and who is located in the hospital.

In both (129) and (130) the explicit marker *además* will be disconnected from the dependency tree of the annotation and added to the syntactic tag of its head right next to the discourse relation name. (131) shows as example the annotation of sentence (129).

(131)	<u>Position</u>	<u>Word form</u>	<u>Head</u>	<u>Syntactic function</u>
	1	Esperaban	0	root
	2	echar	1	dobj
	3	un	4	det
	4	pasodoble	2	dobj
	5	y	–	–
	6	les	7	iobj
	7	cayó	5	coord:enumeration y además
	8	además	–	–
	9	un	10	det
	10	mitin	7	dobj
	11	político	10	amod
	12	.	5	punct

An *enumeration* can also be introduced by inserts starting with participles and gerunds. This can happen at the beginning of a sentence, as shown in (132), but also within (133) or at end position of the sentence (134).

- (132) [**Siguiendo** la línea del Estado jacobino francés,] [Kemal Ataturk y sus seguidores se negaron a aceptar minorías lingüísticas.]

In line with the French Jacobin state, Kemal Ataturk and his followers refused to accept linguistic minorities.

- (133) [El arquitecto Miguel Fisac,] [**nacido** en Daimiel en 1913,] [ha fallecido a primera hora de hoy en su domicilio de Madrid.]

The architect Miguel Fisac, born in Daimiel in 1913, has died early today at his home in Madrid.

- (134) [Los tribunales celebraron otros 19 juicios en los que existía delito,] [**dictando** una sola sentencia absolutoria.]

The court held another 19 trials, in which there was criminal action, and pronounced only one acquittal.

As stated before, also events described in similar structures will be annotated with the *enumeration* relation following the *similarity* relation in Wolf and Gibson (2006) and Hoobs' *parallel* relation (1985).

Nevertheless, these will only be annotated at sentence level and if an explicit lexical marker makes them identifiable. Neither (135) nor (136) do have a lexical marker for *similarity*. While (135) would be annotated with *enumeration* thanks to the coordination *y*, example (136) would not, as the present annotation does not work across sentences.

- (135) Hay un tren la vía 1 y hay otro tren en la vía 2.

There is a train on platform 1 and there is another train on platform 2.

- (136) a. Hay un tren en la vía 1.

There is a train on platform 1.

- b. Hay otro tren en la vía 2.

There is another train on platform 2.

- **Inclusion**

Inclusion gathers several discourse relations used in Hobbs and Wolf and Gibson. This more abstract concept holds between two nodes in which the *satellite* gives more detailed information about the *nucleus*. This can be done by means of a generalization, exemplification or modification of what is said in the *nucleus*. This relation can be marked by means of explicit lexical markers like *por ejemplo* ('for example'), as in (137), but in most cases by inserts like appositions, which give a more detailed view on a noun phrase (138), or time expressions and prepositional phrases which modify the whole sentence (139 and 140).

The latter two cases can be identified by help of their syntactic function and by means of the commas which are part of the insert.

Some examples for explicit cases:

- (137) a. [Dirigentes regionales del PSOE reconocen que] [el censo oficial es irreal] [y admiten haber visto muchas situaciones anómalas].
PSOE regional leaders recognize that the official census is unreal and admit to having seen many anomalous situations.

b. [**Por ejemplo**, se ha dado durante un tiempo el cerrojazo de algunas agrupaciones a admitir militantes].

For example, there has been for a certain time a stop in some groups to admit militants.

- (138) [El laboratorio productor del medicamento,] [**la empresa norteamericana Pfizer,**] [respaldó la medida.]

The laboratory which produces the medicament, the American company Pfizer, supported the measure.

- (139) [**El mes pasado,**] [el desempleo aumentó entre los menores de 25 años en 2.435 personas respecto a julio.]

Last month, unemployment increased among those under 25 years in 2,435 people regarding July.

- (140) [**En Catalunya,**] [la Conselleria de Sanitat ha renunciado a preparar una campaña.]

In Catalonia, the Ministry of Health has resigned to prepare a campaign.

Example (139) shows how a temporal expression can be used to elaborate a sentence by giving background information. In this case *El mes pasado* ('Last month') would be a segment with *mes* as head and marked with the *inclusion* relation. In (140), one can see a prepositional phrase which modifies the whole sentence and not only the main verb. These cases will also be treated as segments which contribute an inclusion relation to the annotation. The comma after the prepositional phrase is a good indicator for their identification.

An example for an implicit case:

- (141) a. Entre los acusados se encuentran, al parecer, parte de la flor y nata de las finanzas alemanas.

Among the accused are apparently part of the cream of German finances.

- b. Se trata de millonarios que usaron fundaciones en Liechtenstein para desviar grandes cantidades a cuentas encubiertas en el Principado alpino, en la vecina Suiza o en otro paraíso fiscal.

They are millionaires who used foundations in Liechtenstein to divert large amounts to undercover accounts in the Principality Alpine, in the neighbour country Switzerland or in another tax haven.

Examples (137) and (141) are not only examples for *inclusion* but show also that the relation does not necessarily hold between two structures in the same sentence. Since discourse is not limited to work at sentence level neither this annotation does. Nevertheless, the annotation identifies the relation, stops at sentence level and leaves the establishment of the relation connection to another sentence to a discourses parser.

In (141), sentence b is an elaboration of sentence a, which is not introduced by an explicit lexical marker and will therefore not be annotated in this work. This task will be left to a full discourse parser.

- **Opposition**

This relation holds between two segments in which an expected consequence of the other segment is absent or one of the segments presents a contrast to the other one. It combines basically *violated expectation* and *contrast* relations, which both Hobbs and Wolf and Gibson use in their works. Both concepts have semantically a point in common, which is the contrariness of their content to what is expected according to the context. This makes them a good fit for a grouping into a superset.

The advantage of a more abstract concept as *opposition* is in this case that the dependency annotation does not have to distinguish between both of them which is not always a trivial decision.

The *opposition* relation is in Spanish normally expressed explicitly by help of conjunctions. Nevertheless, some conjunctions like *pero* ('but') can introduce both a *violated expectation* (142) and a *contrast* (143). The use of *opposition* makes it unnecessary to disambiguate in these cases; a task that will be done by a discourse parser if needed. (144) exemplifies *mientras que* ('while') as lexical marker for an *opposition* relation.

- (142) [Es cierto que el público presiona mucho,] [**pero** todo dependerá del Barça.]

It is correct that the spectators pressure a lot, but all depends on Barça.

- (143) [Es un mundo más rico,] [**pero** es más injusto.]

It is a richer world, but not a fairer one.

- (144) [La niña, de ocho años, es fruto de una relación anterior con Carlos León,] [**mientras que** Rocco, de cinco, es hijo de Ritchie.]

The girl, aged eight, is the result of a previous relationship with Carlos Leon, while Rocco, five, is the son of Ritchie.

This annotation will therefore use the *opposition* relation, which then offers a rich input for a discourse parser.

- **Attribution**

This relation holds between two segments in which one gives a statement and the other indicates the source of the statement. One can possibly find both direct and indirect speech in the segment which includes the statement. The segment which indicates the attribution can be introduced by a prepositional phrase with *según* ('according to') but also by a main clause making use of a series of verbs, such as *decir* ('say'), *preguntar* ('ask') or *contestar* ('respond'), which indicate the attribution.

This relation corresponds to the relation introduced by Wolf and Gibson under the same name. It is in some way different to the other discourse relations as it is not always directly part of the discourse content but gives in most cases information about the source of what is being said in the discourse. See some examples in (145) and (146):

(145) [**Según** los sindicatos,] [las manifestaciones en toda Francia han aglutinado a dos millones.]

According to the syndicates, the demonstrations throughout France have bonded to two million.

(146) [Harper **dijo**] [que Banesto fue intervenido por posibles motivos económicos]

Harper said that Banesto was interfered for possible economic reasons.

- **Causality**

The concept *causality* reflects a relation between segments in which one of the two segments is at least partly responsible for the second to happen. This can be the case of conditions (147 and 148) in which one event has to happen so that the other event can become reality. The concept also includes purposes (149), which give a reason why a certain event happens. Furthermore, both causes and effects are included (150 and 151), which both give arguments for a certain event to happen. They are actually used as one single concept in Wolf and Gibson (2006) as only the *nucleus-satellite* direction reverses between them.

(147) En resumen, hablar del trauma es bueno para las personas que necesitan hacerlo, **siempre que** no se haga de forma reiterada.

In short, talk about the trauma is good for people who need to do so, provided it is not done repeatedly.

(148) **Cuando tengas** el dinero, hablamos del asunto.

When you have the money, we will talk about the topic.

(149) El anteproyecto de la ley de Igualdad recoge numerosas medidas políticas y sociales, **con el objetivo de** evitar la discriminación.

The draft of the equality law gathers several political and social measures in order to avoid discrimination.

(150) La Mars Climate se estrelló en Marte **porque** la NASA no tradujo kilómetros a millas.

Mars Climate smashed against Mars because NASA did not translate kilometers into miles.

(151) Adoro el mundo del cine, **por eso** puedo criticarlo.

I love the movie world, that is why I can criticise it.

Conditions are normally introduced by adverbial clauses with specific conjunctions that state explicitly the condition. Generally speaking, these segments explain a situation which has to hold as true in order to make happen what is stated in the other segment.

In the case of a *purpose*, one of the segments describes the aim of another action. Spanish makes use of adverbial (e.g. *para que...*, ‘so that...’) and infinitival phrases (149) to express this relation.

Both types of relations between segments will be annotated as *causality*. The same treatment will be applied to statements which explain the *cause* (150) or the *effect* (151) of another event to happen. These segments will typically be introduced by a set of conjunctions which serve as marker for this relation.

- **Same**

This relation is established between two segments which would be one single segment if they had not been broken by the insertion of another one in between. The inserted segment elaborates the first segment of the *same* relation. They were separated due to the insertion of another one in between, which elaborates the first one. This can happen if a nominal phrase is separated from its predicate by means of a different segment. Relative clauses and appositions are typical candidates for being introduced in between, as well as other segments which elaborate the previous one, or a fragment of the previous one. The *same* relation is also considered in Wolf and Gibson but not in Hobbs' work. This is probably due to the fact that it is not an actual coherence relation but an epiphenomenon which results from the introduction of a different type of segment within a continuous one (Wolf and Gibson, 2006:28). Here are some examples of its usage:

- (152) [**La empresa,**] [**que** mantendrá su sede en Múnich,] [pretende con la reorganización simplificar su estructura y ahorrar costes.]

The company, which will keep their office in Munich, wants to simplify structures and save costs by the reorganization.

- (153) [La **médico** de empresa,] [**quien** ejerce como tal en el Hospital de Guipúzcoa desde el año 1985,] [asegura que se encuentra 'en una continua sensación de indefensión'.]

The company's doctor, who works as such in the Hospital of Guipuzcoa since 1985, says she is in 'a continuous feeling of helplessness'.

- (154) [La **acusación,**] [**representada** por los letrados José María Loperena y Mateu Seguí,] [intentó demostrar que las 21 personas fallecieron al no poder escapar de Los Pinares.]

The prosecution, represented by the lawyers Jose Maria Loperena and Mateu Segui, intended to show that 21 persons died because they could not escape from Los Pinares.

The annotation will mark the head of the inserted segment with an *inclusion* relation and the head of the first part of the split segment will be marked with a *same* relation. In (152) two nodes would be marked as shown in (155):

(155) La
 empresa nsubj:same
 ,
 que
 mantendrá rmod:inclusion
 su
 sede

6.2 Discourse Parsing Customised Dependency Creation

The following subsections explain the main ideas behind this approach (Subsection 6.2.1) and the linguistic criteria (Subsection 6.2.2) regarding head selection which are chosen for this dependency model adapted to discourse parsing. Then, parsing-related points are discussed (Subsection 6.2.3) and the tagset for Discourse Dependencies is presented (Subsection 6.2.4). Afterwards, the implementation of the system, which annotates the data automatically, is described (Subsection 6.2.5) and the results of the conversion process are presented (Subsection 6.2.6).

6.2.1 Approach

The previously explained set of Discourse Dependencies Relations can now be used to enhance the discourse-adapted dependency model with rich information. The idea is to identify explicit lexical discourse markers, assign them a discourse relation and enhance the corresponding nodes in the dependency annotation with this information. This can be done by means of the syntactic tagset, which will hold the information about the identified discourse relation and about the lexical marker which introduces it. Besides, the linguistic criteria for the head selection in the dependency model adaptation will be

chosen according to the needs of discourse parsing. A constraint on the enhancement of discourse relations is that a node has to be a possible head of a segment in order to add the regarding discourse information.

It is important to comment here that this does not imply a full segmentation of the corpus into discourse segments. It will only be checked by some basic rules if a certain node could be the head of a segment. If this test is positive, then a discourse relation can be added to the node's syntactic function. The test will be based on the segmentation rules indicated in Badia, Saurí and Suñol (2012), as it is a state-of-the-art annotation guide for Spanish.

6.2.2 Linguistic Criteria

This section takes a look at the head selection criteria chosen for Discourse Dependencies. Discourse parsing is interested in the extraction of the events in a text and furthermore on discourse specific information about the way these events are connected with each other in the text. The general implemented linguistic criteria have therefore a focus on semantic information.

Most decisions are similar to those described for Temporal Dependencies (Subsection 5.2.2). The head selection criteria that differ from the previously explained ones for Temporal Dependencies (Chapter 5) are explained in this subsection.

6.2.2.1 Coordinations

Coordinations will be treated differently compared to both the syntactic and the temporal model. The reason is mainly that discourse relations can hold between individual parts of a coordination structure. This happens in cases of oppositions (156 and 157) and enumerations (158 and 159).

(156) Se planteó comer hormigas, **pero** no pudo.

He considered to eat ants, but he couldn't do it.

(157) a. La PET reúne lo más avanzado de la física.

The PET gathers the most advanced parts of physics.

b. **Pero** su utilización está siendo frenada por el alto precio de la prueba.

But its usage is slowed by the high price of the test.

(158) Su pasión estalló durante un crucero por el Mediterráneo y duró hasta el fin de sus días.

His passion exploded during a cruise on the Mediterranean and lasted until the end of his life.

(159) a. El agente ha sido trasladado al hospital comarca, donde se encuentra en estado crítico.

The agent has been transferred to the regional hospital, where he is in critical condition.

b. **Además**, hay otra persona herida en el mismo suceso y que está ingresada en el hospital.

Besides, there is another person who was injured in the same incident and who is located in the hospital.

(157) and (159) are examples which show discourse relations between two sentences, while those relations in (156) and (158) hold within the sentence. If the conjunction in (156) and (158) was the head of the coordination structure, then this would imply a disadvantage for a discourse parser. The parser would then have to check if an *enumeration* or *opposition* relation marked node is the head of a coordination in order to know if the indicated relation holds at sentence level or possibly between two sentences. Therefore coordinations will be treated differently in the discourse dependencies model.

The head of the coordination structure will be the head of the first segment and the other segments connect in a chain structure to the closest previous segment of the coordination structure.

In (158), *estalló* ('exploded') will be the head of the coordination structure (and of the sentence in this case) and *duró* ('lasted') would be its dependent, which will have

marked *enumeration* as discourse relation. Furthermore, the conjunction is disconnected from the dependency tree and added as lexical marker for the relation to *duró* (160).

(160) estalló root
 duró coord:enumeration|y

The dependency function indicates then that *duró* is in an *enumeration* relation to *estalló* and that this relation works at sentence level. In (159b), *Además* (‘Besides’) is a lexical marker for an *enumeration* between sentences (159b-159a) and is marked as such since the root node of the sentence will be enhanced with this information. The discourse model facilitates in this way the work of a discourse parser.

The treatment of coordinations between non-segment structures (as in 161) is also affected by this change since the treatment of coordination structures shall be coherent.

(161) Alemania y Japón proponen una reforma profunda.
 Germany and Japan propose a thorough reform.

(162)	<u>Position</u>	<u>Word form</u>	<u>Head</u>	<u>Syntactic function</u>
	1	Alemania	4	nsubj
	2	y	1	conj
	3	Japón	1	coord
	4	proponen	ROOT	root
	...			

Example (162) shows the treatment of a coordination in a noun phrase in which both the conjunct and the second noun have the first noun as head. In a coordination of three nouns, the third noun would have the second noun as its head in this chain-like treatment. The conjunct is not attached to the syntactic function in these cases as it does not introduce an *enumeration* relation.

6.2.2.2 Complementisers

It was decided to treat complementisers, which do not serve as explicit lexical markers for discourse relations, slightly different than in Temporal Dependencies.

While in the temporal model they were marked with a *compl* function and attached to the semantic verb they introduced, in the discourse model annotation they will be disconnected from the dependency tree in order to optimise path length in the annotation even more.

(163) Harper dijo **que** Banesto fue intervenido.

Harper said that Banesto was interfered.

(164) que compl

(165) que –

(164) shows the annotation of the complementiser *que* ('that') in sentence (163) according to Temporal Dependencies, while (165) indicates how the annotation in Discourse Dependencies is conducted. Its head in Temporal Dependencies would be *intervenido* ('interfered'), while it is left out of the dependency annotation in Discourse Dependencies.

6.2.3 Processing-related Assumptions

This section clarifies a couple of points regarding the processing of the data within the automatic adaptation to Discourse Dependencies.

6.2.3.1 Several Discourse Relations at One Head Node

In a dependency annotation of this model, it is possible that a node holds more than one discourse relation at its syntactic function tag. This happens as both the node itself and its surroundings can introduce discourse relations. In most cases only one discourse relation will be present at one node, but it should be taken into account that it is possible

for one single node to hold more than one identified relation. This can happen because identified relations are always attached to the head node of a dependency structure. Thus, a node can be head of several nodes which evoke a discourse relation. Furthermore, the head node itself can evoke a relation which will be attached to it. In (166), for example, the lexical marker *Asimismo* ('Furthermore') evokes an enumeration relation, which will be attached to its head, and its head *apuntaron* ('emphasised') itself also evokes an attribution relation. Both relations will then be attached to the head node, as shown in (167).

(166) Los responsables en urología y psiquiatría subrayaron que las dosis de la terapia deben individualizarse. **Asimismo apuntaron que** no es significativa su mejoría.

The responsible people in urology and psychiatry emphasized that the dose of the therapy has to be individualised. Furthermore, they pointed out that the improvement is not significant.

In this case *Asimismo* evokes in the second sentence an *enumeration* relation and *apuntaron* an *attribution* relation. Both will be attached to the head *apuntaron*.

(167) apuntaron root:attribution:enumeration|Asimismo

6.2.3.2 Disconnection of Nodes in the Annotation

Nodes will be disconnected from the dependency annotation if they introduce a discourse relation explicitly and are not the segment's head. This can happen in case of conjunctions and prepositions, as in the Temporal Dependencies annotation, but also for verb modifiers, such as adverbs (e.g. *generalmente*, 'generally'). (168) exemplifies the case of a coordination (*pero*, 'but'). There are two clauses in the example, which are connected by a conjunction (*pero*). This conjunction is a lexical marker for an *opposition* relation. The conjunction will be disconnected from the dependency tree and

the relation name plus word form will be added to the syntactic tag of its head. The head of the second clause (*quería*, ‘wanted’) holds this information and indicates in this way its *opposition* relation to the head of the first clause (*Era*, ‘It was’).

(168)	1	Era	0	root		vsii3s0	ser	
	2	una	3	det		di0fs0	uno	
	3	incomodidad	1		attr	ncfs000	incomodidad	
	4	pero	_	_	cc		pero	
	5	su	6	poss		dp3cs0	su	
	6	hermano	8	nsubj		ncms000	hermano	
	7	no	8	neg		rn	no	
	8	quería	1		coord:opposition pero	vmii3s0	querer	
	9	comprarle	8		dobj	vmn0000	comprar	
	10	un	11	det		di0ms0	uno	
	11	aparato	9	dobj		ncms000	aparato	
	12	nuevo	11	amod		aq0ms0	nuevo	
	13	.	4	punct		fp	.	

If a node is disconnected, its head adds the following information to its syntactic function as seen in (167) and (168):

:relation_name|deleted_word(s)

The following example shows a node which introduces a discourse relation but serves as head in the Discourse Dependencies annotation and which therefore will not be disconnected (169).

(169) Pedro **dijo** que había comprado el agua.

Pedro said that he had bought water.

(170) dijo root:attribution

As seen in (170), only the discourse relation name is added to the syntactic function in this case.

6.2.4 Tagset for Discourse Dependencies

The tagset (Table 25) is an enhanced version of the SSD dependencies tagset. The syntactic adaptation to linguistic criteria based on the needs of discourse parsing implied several changes in the head selection of the dependency annotation. Due to these head changes, the introduction of seven new tags was necessary. Three of the tags were introduced due to the focus on semantic verb information. These three new tags are *aux*, *asp* and *modal* and give detailed information about the type of auxiliary verb that is connected to the main verb which contains the semantic information. Three additional tags are conditioned by the treatment of adverbial clauses and oblique objects as in them the verb carrying the lexical semantic information is now head of the corresponding structure. The tags correspond to the general tag *mark*, and the two finer-grained tags *markobl* and *markadvl*. The underscore _ was added to mark nodes which are disconnected from the dependency annotation.

SSD Tag	Discourse Dependencies Tag	Full name
root	root	root
dep	dep	dependent
arg	arg	argument
	aux	auxiliar
	asp	aspectual verb
	modal	modal verb
comp	comp	complement
attr	attr	attributive
cpred	cpred	predicative complement
obj	obj	object
cobj	cobj	complementizer object
dobj	dobj	direct object
iobj	iobj	indirect object
oobj	oobj	oblique object
pobj	pobj	object of a preposition
vobj	vobj	object of verb
crobj	crobj	object of comparative
	mark	marker
	markobl	oblique marker
	markadvl	adverbial marker
subj	subj	subject
nsubj	nsubj	nominal subject
csubj	csubj	clausal subject
coord	coord	coordination
conj	conj	conjunct

agent	agent	agent
reflec	reflec	reflexive (“se”)
te	te	textual element
mod	mod	modifier
abbrev	abbrev	abbreviation modifier
amod	amod	adjectival modifier
appos	appos	appositional modifier
advcl	advcl	adverbial clause modifier
det	det	determiner
infmod	infmod	infinitival modifier
partmod	partmod	participial modifier
advmod	advmod	adverbial modifier
neg	neg	negation modifier
rcmod	rcmod	relative clause modifier
nn	nn	noun compound modifier
tmod	tmod	temporal modifier
num	num	numeric modifier
prep	prep	prepositional modifier
prepv	prepv	prep. mod. of a verb
prepn	prepn	prep. mod. of a noun
prepa	prepa	prep. mod. of adjective
poss	poss	possession modifier
punct	punct	punctuation
	–	disconnected from the dependency structure

Table 25: Discourse Dependencies Tagset

While the total number of tags climbs up to 49, it has to be taken into account that several tags are not actually used in a dependency annotation. Some tags serve merely as general tag in case no finer-grained tag can be chosen (such as *mark*), other tags have not been used by the algorithm (such as *infmod*). In total, there were 42 tags used for the actual annotation of AnCora Discourse Dependencies.

The following list gathers all not used tags in the annotation:

arg	subj	nn
comp	mod	
obj	infmod	

Table 26: Tags without usage in AnCora Discourse Dependencies

6.2.5 Implementation

This section presents the implemented algorithm and describes the used lexical markers and patterns for each relation of DDR.

6.2.5.1 Creation of AnCora Discourse Dependencies

The implementation of the discourse model is straightforward as it can be seen in the algorithm (Figure 24).

```
1 function DISCOURSE_DEPENDENCY_ANNOTATION(ssd_corpus):  
2 for ssd_sentence in ssd_corpus:  
3     disc_sentence=general_adaptations(ssd_sentence)  
4     disc_sentence= identify_discourse_relations(disc_sentence)  
5     write_discourse_dependency_annotation(disc_sentence)
```

Figure 24: Discourse Dependencies algorithm

First SSD dependencies are read sentence per sentence (line 2). Then each sentence is automatically adapted to the discourse model. This implies a general adaptation (line 3) of the linguistic criteria to this specific model, such as semantic verb focus, coordination treatment, etc. The treatment is comparable to the one presented for Temporal Dependencies. The algorithm uses for this task 18 rules which make use of lemma, head and part-of-speech information in order to conduct the adaptation. The rules work in a way that if a certain pattern is detected, the respective structure gets a new head according to the chosen linguistic criteria, the syntactic function is adapted to its new use and other dependents get an updated head number in order to connect to the new head.

Afterwards discourse relations are identified (line 4) and the appropriate treatment for this model is applied. This is done mostly by the use of lexical markers (see next subsection) which identify the different relations. 66 rules were implemented which make use of the lexical markers to identify patterns for specific relations. These rules adapt the head selection then and add the identified information to the syntactic function of the structure's head. The head and part-of-speech information is additionally used in

order to check if a node could be the head of a segment since this is decisive for the applied treatment. Finally, AnCorra Discourse Dependencies are annotated (line 5).

6.2.5.2 Lexical Markers and Linguistic Patterns

As explained in the previous section, the algorithm made use of 18 rules for the adaptation to head selection criteria in this model and applied 66 rules for the identification of discourse relations and the respective adaptation of the implied structures to the chosen linguistic criteria. The following subsections provide information about the detected lexical markers and linguistic patterns for each relation.

- **Attribution**

The following two tables show the detected lexical markers for *attribution*. One preposition was identified as lexical marker (Table 27) and 33 verbs (Table 28).

según (according to)

Table 27: Prepositions as lexical marker for attribution

preguntar (ask)	indicar (que) (indicate [that])	recordar (remember)	reiterar (repeat)	reafirmar (reaffirm)
decir (say)	comentar (comment)	pedir (que) (ask [to])	quejar (complain)	replica (reply)
susurrar (whisper)	informar (inform)	destacar (single out)	confiar (trust)	matizar (emphasize)
murmurar (whisper, purr)	responder (respond)	sostener (claim)	subrayar (que) (emphasize [that])	admitir (admit)
apuntar (que) (note)	considerar (consider)	señalar (que) (specify [that])	revelar (reveal)	contester (respond)
confirmar (confirm)	explicar (explain)	asegurar (assure)	creer (think)	
declarar (declare)	reconocer (que) (recognize [that])	concluir (conclude)	afirmar (confirm)	

Table 28: Verbs as lexical markers for attribution

As Spanish is a rich language in vocabulary, there are also many options to express an *attribution* relation by means of verbs. The list has been created by observing the corpus and by consulting several dictionaries. Nevertheless, the list does not claim to be complete and could be expanded with a further investigation. But the most common lexical markers are included in this version. Note that there are several lexical markers which have a different meaning according to the expression they connect to. As indicated in the previous table, some verbs only introduce *attribution* if they connect to *que* ('that'), as shown in (171 and 172).

(171) a. Dirigentes regionales del PSOE **reconocen que** el censo oficial es irreal.

a. PSOE regional leaders recognise that the official census is unreal.

b. El bebe **reconoce** a su madre.

b. The baby recognises her mother.

(172) a. El presidente **pide que** se mantenga la calma.

a. The president asks to stay calm.

b. El chico **pide** dinero en la calle.

b. The boy asks for money on the street.

Another restriction was also implemented. The use of impersonal forms does not introduce an *attribution* relation, as there is no direct source given (see 173).

(173) a. En Alemania **se dice** que la tercera es la vencida.

a. In Germany it is said that all good things come in threes.

b. **Pablo dice que** llegará tarde.

b. Pablo says that he will arrive late.

While (173b) does introduce an *attribution* relation, (173a) does not give a direct source.

- **Enumeration**

This relation can be introduced in several ways. Coordinations and adverbial and infinitival clauses cover most of them.

The model treats Spanish coordinations with the restriction that only coordinations between non infinitival verbs introduce an *enumeration*. Coordinations between other structures (such as nominal phrases) are not considered as segment and therefore not treated as this relation.

Regarding adverbial clauses it can be said that several conjunctions can introduce an enumeration. The implemented lexical markers can be seen in the following table:

antes de que (before)
después de que (after)
mientras que (while)
cuando + indicative (when)

Table 29: Adverbial clause introducing conjunctions for *enumeration*

Furthermore the following lexical markers which introduce infinitival clauses have been treated by the algorithm:

antes de (before)
después de (after)
tras (after)

Table 30: Prepositions introducing infinitival clauses for *enumeration*

Note that subordinated clauses can also have an infinitival verb as head, and introduce an *enumeration* if the head of the other clause is a non-infinitival verb. This can be seen in example (174).

- (174) Ya se permitían lujos innovadores **antes de** existir la nueva ley.
 Innovative luxuries were already allowed before the existence of the new law.

Also a selection of adverbial modifiers are lexical markers for an *enumeration*:

antes (before)	luego (later)	además (additionally)
después (after)	pronto (soon)	asimismo (additionally)
ahora (now)	simultáneamente (simultaneously)	también (also)
primero (first)		

Table 31: Adverbial modifiers for enumeration

- **Inclusion**

The *inclusion* relation gathers several fine-grained relation concepts such as exemplification, background or elaboration, which are sometimes hard to identify by means of lexical markers. The implemented lexical markers can be seen in Table 32 and the implemented patterns in Table 33:

por ejemplo (for example)	respectivo a (regarding)
ahora que (now that)	

Table 32: Lexical markers for inclusion

, como + noun
(, such as + noun)
temporal modifiers

Table 33: Patterns for inclusion

The first rule in Table 33 basically says that *como* (‘such as’) converts into a lexical marker if it is followed by a noun and preceded by a comma. This case corresponds to insertions which serve as exemplifications of noun phrases such as in (175).

(175) Los culpables, **como el entrenador**, justificaron después la pérdida.

The culprits, such as the coach, justified later the defeat.

The second rule treats temporal modifiers according to the annotation guidelines (Badia, Saurí, Suñol, 2012). It has to be commented here that this rule relies only on the syntactic function tag *tmod* in SSD dependencies. The algorithm for Discourse Dependencies does not imply a time expression identification, such as implemented in the algorithm for Temporal Dependencies. Example (176) shows a time expression which would be captured by this rule.

(176) **El mes pasado**, el desempleo aumentó entre los menores de 25 años en 2.435 personas respecto a julio.

Last month, unemployment increased among those under 25 years in 2,435 people regarding July.

As explained in the annotation guidelines, cases connected to impersonal constructions or the verb *haber* do not introduce an *inclusion*.

• Opposition

The following table covers all the implemented lexical markers for *opposition*, which are in this case all conjunctions:

aunque (even if)	sino que (but)
sin embargo (nevertheless)	por más que (however much)
a pesar de ello (nevertheless)	pero (but)
pese a que (despite)	aún así (nonetheless)
mientras que (while)	no obstante (nonetheless)

Table 34: Lexical markers for opposition

The conjunctions can introduce subordination, as for example *aunque* (‘even if’), but also coordination, as seen in *pero* (‘but’).

• Causality

The concept of *causality* gathers all relations in discourse in which one of the two segments is at least partly responsible for the second to happen. As explained in Subsection 6.1.2.2, this includes conditions, purposes, causes and effects. The following tables present all the lexical markers that were included in the implementation for *causality*.

Spanish makes use of several conjunctions which introduce a condition (Table 35), a cause (Table 36), an effect (Table 37) or purpose (Table 38). All of them are lexical markers to identify the *causality* relation.

siempre que (provided that)	hasta que (until)
si (if)	supuesto que (provided that)
siempre y cuando (provided that)	cuando (+ subjunctive) (when)

Table 35: Conjunctions for causality (conditions)

porque (because)	a causa de que (because)
ya que (as)	debido a que (because)
dado que (as)	por más que (however much)
puesto que (as)	

Table 36: Conjunctions for causality (cause)

por lo tanto (therefore)	de modo que (so that)
por eso (therefore)	de manera que (so that)
conque (hence)	por tanto (therefore)
tal que (so that)	por consiguiente (therefore)
así pues (hence)	tanto que (therefore)
de esta manera (so that)	así que (therefore)
de este modo (so that)	

Table 37: Conjunctions of causality (effect)

para que (so that)
a fin de que (thereby)
con el propósito de que (with the aim to)
con el fin de que (with the aim to)
con (el) objeto de que (with the aim to)
con (el) objetivo de que (with the aim to)

Table 38: Conjunctions of causality (purpose)

But not only conjunctions were identified as lexical markers for *causality*. Several prepositional structures can also introduce this relation.

In the case of a purpose, one of the segments describes the aim of a previous action. Spanish makes use of adverbial (Table 38) and infinitival phrases (Table 39) to express this relation. Examples can be seen in the following table:

para (for)
a fin de (in order to)
con el objetivo de (with the aim to)

Table 39: Prepositional structures for causality (purpose)

Additionally, the preposition *por* is marked as lexical marker when connected to an infinitival clause, as it introduces the cause of a happening.

por + infinitive (for)

Table 40: Prepositional structures for causality (cause)

- **Same**

The *same* relation does not seem to be identifiable by lexical markers. Two rules were implemented in order to solve this task.

The first rule treats appositions by means of the part-of-speech tag and searches for punctuation between the noun and its apposition in order to check if they are in separated segments. The second rule checks if segments are split due to relative clauses. If this is the case, again a *same* relation is introduced.

(177) **Joaquín Prats**, director del departamento, **asegura** que es injusto.

Joaquin Prats, director of the department, assures that it is unfair.

(178) **La hamburguesa**, que cuesta 1 euro en Barcelona, **supera** los 2 euros en
Berlín.

The hamburger, which costs 1 euro in Barcelona, exceeds 2 euros in Berlín.

6.2.6 Customisation Results

This section has been divided into five subsections. First, the tasks, which are involved in the evaluation process, and the criteria for a correct annotation are presented (Subsection 6.2.6.1). Then, the evaluation corpus is explained (Subsection 6.2.6.2) and the evaluation results are discussed (Subsection 6.2.6.3). Afterwards, several problems are explained that came up in the automatic annotation (Subsection 6.2.6.4) and then some final words about AnCora Discourse Dependencies are given (Subsection 6.2.6.5).

6.2.6.1 Evaluation Tasks

The creation of the discourse model can be mainly divided into two tasks. The first is the identification of discourse relations between possible segments and the second is the general adaptation of the dependency annotation according to the chosen criteria for this discourse annotation. Both individual tasks shall therefore be evaluated in this section.

The identification of discourse relations itself implies several steps, since the adaptation program has to decide if a candidate is a possible segment head (only those are marked with a discourse relation), and since it also has to select the correct relation from the chosen set.

Discourse Dependencies Relations gather a total of six relations, which are added to the syntactic function tag of the candidate by means of a colon and the name of the corresponding relation. See, for example, (179).

(179) *dijo* root:**attribution**

If there is also a specific lexical marker which introduces a discourse relation, and which is not itself the head of the segment, this marker (node) is dropped from the discourse annotation and added also to the syntactic function tag by means of a vertical line and the word form of the lexical marker, as in (180).

(180) *fue* dobj:**causality|porque**

A correct annotation has to include the mentioned elements. The discourse relation identification task of this evaluation implies the choice of the correct node and the correct discourse relation. The dropping and addition of lexical markers which introduce a specific relation is counted towards the general adaptation of the dependency annotation.

Thus, (180) without "|porque" would be counted as a correct identification of the discourse relation but would count as an error for the syntactic function tag within the results of the general adaptation of the model.

6.2.6.2 Evaluation Corpus

The automatic annotation was done for the whole AnCora corpus and evaluated for the same 21 files which were selected for the evaluation of Time Dependencies. This was done by two professional linguists, which annotated each half of the evaluation corpus. Additionally, complex structures were discussed between both of them.

6.2.6.3 Result Data

The evaluation was divided into two individual tasks: the identification of Discourse Dependencies Relations and the general adaptation to the task-specific linguistic criteria. The results are presented in this subsection in the same way and take first of all a look on the DDR identification.

The 21 files of the evaluation corpus contain 319 discourse relation instances which had to be correctly identified. The automatic annotation identified 262 of those 319 instances which corresponds to a recall value of 0.82. This seems to be a good result considering the number of available discourse relations and the fact that also segment boundaries have to be considered. The precision value of 0.85 shows also that the implemented rules were carefully chosen and did manage in this way to not only identify a high amount of discourse relations correctly but to maintain also a high quality in the annotation.

	Precision	Recall	F-Measure⁷
Evaluation corpus	0.85	0.82	0.83

Table 41: Discourse relation identification results

The next task of the automatic annotation was the general adaptation of the model according to the established criteria. This implies on the one hand direct changes like the addition of discourse relations to the syntactic function tags of certain nodes, but on

⁷ F-Measure=(2*precision*recall)/(precision+recall)

the other hand also changes to the surroundings of those nodes. An important change in comparison to the temporal model has been done regarding coordination structures since a discourse relation can hold within those structures but does not necessarily have to.

The general adaptation has been measured in terms of Labelled Attachment Score (LAS), which implies the correct value for both head and syntactic function label, Unlabelled Attachment Score (UAS), which measures only the correctness of heads, and, finally, with Label Accuracy (LA), which measures correct syntactic label values.

	UAS	LA	LAS
Evaluation corpus	0. 82	0. 78	0. 76

Table 42: General adaptation results

As Table 42 shows, the results are really positive considering that errors in the input data (AnCora Surface Syntax Dependencies) will most probably lead to errors in the annotation done by the discourse model, and sometimes affect also surrounding nodes. In the evaluation data, 114 of the 281 wrong heads in the model adaptation arise from former errors in the input data. A similar situation can be observed for the syntactic functions, as 145 out of the 344 errors come up in the same way. Note that one error in the input data can lead to several consecutive errors in the model adaptation and that the selection of a wrong head normally leads to also to a wrong syntactic function. Besides, unspecified syntactic functions (*dep*) were also counted as wrong and those are also taken directly from the input data.

6.2.6.4 Error Analysis

As to be expected in an automatic annotation, not everything was perfect. The following analysis sheds light on the errors that were committed by the automatic annotation. It is divided into problems regarding the head selection and the identification of discourse relations.

- **Head Selection**

The algorithm for the discourse model had problems with complex coordination structures such as *Dado que... y que ...* (181), irrelevant commas which were not dropped from the annotation even if their whole related content was so (182) and short non-verbal structures which were not caught by the implemented rules (183).

- (181) **Dado que** los farmacéuticos trabajan con un margen de beneficio del 29,7% sobre el precio de venta al público **y que** existe un mercado potencial de más de un millón de consumidores, la posibilidad de negocio es enorme.

Since pharmacists work with a profit margin of 29.7% on the selling price to the public and since there is a potential market of more than one million consumers, the business opportunity is huge.

- (182) **Asimismo**, también podrían contribuir a esa decisión las aseveraciones del jefe.

Additionally, they could also contribute to this decision the statements of the boss.

- (183) Aragón dice **que no**.

Aragon says “no”.

In (181), the algorithm was not able to handle correctly the complex coordination structure of the subordinate clause and connected the second verbal phrase to the conjunction *y* (‘and’) instead of the first verbal phrase. The implementation of more coordination-specific rules would make it possible to handle this and similar problems.

In (182), the lexical discourse marker *Asimismo* (‘Furthermore’) was disconnected from the dependency structure, but the comma which accompanied the lexical marker was not. A rule has to be implemented to treat punctuation which is related to disconnected nodes in the annotation.

In (183), the implemented rules had problems to handle the subordinate clause as it does not include a verb. Finer-grained rules could handle these structures.

As seen, the head selection errors are strongly related to the graininess of the implemented rule set which handles the model adaptation. While most structures are already handled correctly, it is still possible to further improve the model adaptation.

- **Identification of Discourse Dependencies Relations**

The identification of discourse relations shows issues with the distinction of prepositional phrases that work as modifier of the whole sentence (184), and which are set as *inclusion*, and those which do not specify the whole sentence (185) or which do not fulfil the content-based criteria according to the annotation guide in Badia, Saurí, Suñol (2012). The correspondent rule would need more information about the content of the prepositional phrase in order to decide if the phrase modifies the whole sentence or not.

(184) **En Catalunya**, la Conselleria de Sanitat ha renunciado a preparar una campaña.

In Catalonia, the Health Ministry has resigned to propose a campaign.

(185) **Entre otras indicaciones**, el departamento destaca que la píldora está contraindicada.

Among other indications, the department emphasizes that the pill is contraindicated.

The discourse relation *attribution* showed already some problems, as it is not a trivial task to decide which lexical markers evoke this relation and in which contexts. The implemented rules did not take into account that in negated contexts and in questions verbs like *creer* ('think') do not evoke this relation (186). The implementation of further specific rules could handle this problem.

(186) ¿Por qué cree que debíamos publicarlas?

Why does he think that we had to publish them?

But not only the addition of further rules can improve the obtained results. Lexical markers are the main factor for a successful DDR identification. The created program has access to a wide range of lexical markers already, but this list can still be expanded. For example, the list of *attribution* related verbs does so far not cover all possible entries. Additionally, there are some verbs which clearly introduce this relation (such as *decir*, ‘say’) but also verbs which are ambiguous in this case. If one compares (187) with (188), only the former introduces an *attribution* relation.

(187) El presidente **pide que** mantenga la calma.

The president asked to stay calm.

(188) El chico **pide** 3 euros.

The boy asks for 3 euros.

Thus, sometimes a small difference distinguishes between the introduction of this relation and a different usage of candidate verbs.

Another problem came up with the preposition *según* (‘according to’), while correctly recognised as lexical marker for an *attribution* relation in cases like (189), it is not correct in other cases like (190), where it does not introduce a source of what is said but refers to a type of virus out of a variety.

(189) El beneficio alcanzó los 2.124 millones, **según** informó la entidad.

The profit reached 2.124 million, according to the bank.

(190) **Según** el tipo de virus que le haya invadido, los anticuerpos pueden sobrevivir durante varios años.

Depending on the type of virus that has invaded him, antibodies survive for several years.

The *enumeration* relation brought up problems in more complex coordinations as those were not always correctly adapted as previously mentioned (181).

Other lexical markers as *así* ('in this way') introduce only a discourse relation in certain positions in the sentence, which was not always handled correctly by the automatic annotation.

(191) a. Los impuestos siguen subiendo.

Taxes are rising.

b. **Así**, es difícil que se recupere la economía.

In this way, it is difficult for the economy to recover.

(192) Su hermano solía comportarse **así**.

His brother usually behaved in this way.

While in a sentence sequence like in (191) the word *así* is a lexical marker which expresses a *causality* relation, this is not always the case for this word. (192) shows a sentence of the evaluation corpus data, which was erroneously identified as *causality* in the automatic annotation. A rule has to be implemented which states this limitation.

Errors related to the non-identification of discourse relations (*false negatives*) come up with the design of the implemented rules. As those rules are handcrafted, only entries that were considered in the creation process can be found within the automatic annotation. Nevertheless, most lexical markers were found, but *tal como* ('such as') is an example for a *false negative* which was not considered in the implemented rules.

The error analysis has shown that the automatic model adaptation can still be improved. The implementation of further lexical markers and rules should result in a higher identification of DDR and less errors in the automatic head selection within the adaptation process.

6.2.6.5 Conclusion

The creation of a discourse-related dependency model has been successful. Discourse Dependencies represents a dependency model which is adapted to the needs of discourse parsing and offers a rich linguistic input. The linguistic criteria on which the model is based correspond to the needs of this NLP task and its implementation makes the relevant information easier to reach for discourse parsers. Furthermore, Discourse Dependencies Relations have been designed as an “abstract” form of discourse relations and enrich the syntactic tags of the model by discourse-specific information.

The resulting adaptation program has already seen a practical use in the automatic annotation of AnCora Discourse Dependencies. The annotation has shown good results and offers an optimised version of AnCora dependencies regarding its application in discourse parsing.

The evaluation of the automatic adaptation has shown that there is still room for smaller improvements, which is in the nature of a rule-based approach, but, generally speaking, the results have been positive. It has to be taken into account that the input data (AnCora Surface Syntax Dependencies) is also the result of an automatic conversion. Despite the fact that the input data resulted already from an automatic conversion, the evaluation results show that the quality of the AnCora Discourse Dependencies annotation is high enough to offer a useful resource for future linguistic projects. Furthermore, it presents a unique proposal for a discourse-optimised dependency annotation and features a set of “abstract” discourse relations, which can be converted into finer grained discourse relations by a discourse parser, if it is convenient for a specific application.

7. Evaluation

Chapter 7 explains the evaluation of the *optimisation* process of the two created dependency models. This includes the use of several standard network analysis measures within a language network environment. The term *optimisation* is used throughout this chapter referring to a measurable improvement of a network's performance, as in Newman (2010:541-551).

7.1 Motivation

Within the previous work of the present research different dependency models have been designed, and been then used to annotate task-based versions of the AnCora corpus. The different models have shown that there is not a unique correct dependency annotation, but that the annotation can be adapted to the further use of the data. Similar observations can also be seen in de Marneffe and Manning (2008) and Silveira and Manning (2015). The models have been adapted having a specific use in mind, so that their structures are optimised accordingly.

In Subsections 5.2.4 and 6.2.6, the conversion process from AnCora Surface Syntax Dependencies to AnCora Temporal and Discourse Dependencies has been evaluated already by comparing the obtained results in each of them with a manually annotated subset of AnCora according to the respective linguistic criteria. This way, we were able to see how well the conversion processes performed in each case and what sorts of errors occurred in each of them. What was still missing is an evaluation on whether the conversion resulted in really optimised models which can facilitate the use of the annotated data in temporal and discourse parsing respectively. As discussed in Section 7.2, this has been done by first building language networks from the dependency annotated data and then by exploring the properties that the resulting networks have. Ultimately this evaluation will tell us whether the newly annotated data result in optimised networks with respect to the original surface-syntax dependency annotation.

Complex networks have been used in language research in different areas (Section 3.3), including syntactic dependencies. Nevertheless, so far this use has been limited to research on general characteristics of the whole networks, as in Ferrer i Cancho et al. (2004). The present evaluation has the advantage to have access to comparable results, since the different models have been used to annotate the same corpus, and can offer a deeper analysis of what is happening inside the language networks.

The idea is therefore to present the general characteristics of the different language networks and, primarily, to have a closer look at individual nodes inside the networks which correspond to relevant nodes according to the linguistic criteria chosen for a specific dependency model. The *optimisation*, which was performed in the adaptation process, should then be reflected in the comparison between the different networks, that is, between the network built from the surface-syntax dependency data on the one side and the networks built from the temporal and discourse dependencies data on the other.

7.2. Evaluation Description

This section describes the hypothesis behind the evaluation and the setup that was used to conduct it. Afterwards, the results are presented and discussed in order to reach conclusions regarding the present work.

7.2.1 Hypothesis

The three versions of the AnCora corpus, which have been created within this doctoral thesis, will be used to create different network representations in order to evaluate their optimisation regarding specific points of interest according to the task-based model.

As described in Chapters 5 and 6, specific adaptations have been implemented into the syntax-based model in order to create a task-oriented version of the base annotation for temporal and discourse parsing.

The nodes that were identified according to the linguistic criteria as specifically relevant for each version should therefore show their *importance* in a network analysis and it is expected that a comparison of their *importance* between the three networks will show how this *importance* has been gained through the changes implemented in the *optimisation* process.

The term *importance* refers throughout this chapter to the measurable results in a network analysis in contrast to the term *relevance*, which will be used to refer to linguistic criteria.

While auxiliary verbs are often attributed a high relevance in a syntax-based annotation, since they contain information about tense, person, number and voice, they should be less *relevant* in the temporal and discourse annotations as they do not contain the semantic information of the verbal construction. On the other hand, time expressions are *relevant* for the temporal model, while conjunctions like *porque* ('because') are essential for the discourse model.

Therefore, a list of *relevant* nodes will be created for each model and their *importance* in the networks will be compared to each other. The network analysis tool *igraph* (Csárdi and Nepusz, 2006) is used to calculate the results.

7.2.2 Network Construction

The networks are constructed as weighted directed graphs by means of dependent-head pairs of the three corpora. Figure 25 shows an extract of the SSD corpus.

ruptura suponer 2
ruptura venir 1
rumor difundir 1
rumor entre 1
rumor o 1
rumor circular 1
rumor comenzar 2
rumor descartar 1

Figure 25: SSD corpus example for network construction

In a network, a dependent-head pair represents two vertices which are connected by an edge which can be walked only in one direction (from dependent to head). The weight of each edge expresses the frequency of how many times this relation exists in the corpus.

The frequency is useful, since a dependent-head pair which occurs in the corpus 1,000 times should have a higher influence on the network composition than one which occurs only 3 times.

7.2.3 List of Relevant Nodes for Evaluation

It was decided to create two lists of lemmas for the evaluation of the optimisation process for the two task-based models. Each list corresponds to candidate nodes which should be relevant for the specific model and show their increase of importance in a network comparison between the base model and the task-based models.

The criteria for the selection of the lemmas took into account the criteria that informed the linguistic decisions for the creation of each model. Both models are designed for information extraction purposes and rely on the semantic information conveyed by events in the sentences. So both lists have in common that they include all verbs which convey semantic information. A list of these nodes was automatically extracted by means of the part of speech sequence *vm* (main verb), and all auxiliary, aspectual and modal verbs were excluded from the list since they normally do not convey the semantically relevant information. The list contains 2,455 lemmas and is included in the evaluation list for both models.

Additionally, it was decided to also add to the list lemmas corresponding to lexical markers in each model. The temporal model identifies a specific type of relevant nodes, which are time expressions (e.g. *hoy*, ‘today’). The lemmas found in time expressions were therefore added to the former list of verbs and the complete list for the evaluation of the temporal model gathers 2,824 lemmas.

On the other hand, the addition of lexical markers for the discourse model could not be implemented. As seen in Chapter 6, the treatment of the lexical markers was different than in the temporal model. The lexical markers for the discourse model which introduce a discourse relation (e.g. *porque*, ‘because’) were disconnected from the dependency structures, and their word form and discourse related information has been added already to the syntactic function tag of the corresponding head (Subsection 6.2.3). Those heads are normally the verbs that are already on the evaluation list. Therefore, no further lemmas were added to the evaluation list of the discourse model.

Both lists offer now a good selection of lemmas to show the changes of importance within a network regarding nodes which are considered relevant for the specific model. In this way, the optimisation process from the base model into a task-specific model can be measured.

7.2.4 Measures

A detailed description of the different measures and their significance has already been given in Subsection 3.3.4. This evaluation makes use of three measures regarding general characteristics of the different networks and six node-specific measures. An overview is given in Table 43 and Table 44.

<u>Measure</u>	<u>Summary</u>
Average Path Length	The average number of steps along the shortest paths for all possible pairs of network nodes
Transitivity	Clustering tendency of nodes in the network
Assortativity	Tendency of nodes to connect to nodes with a similar degree to their own

Table 43: General network measures

<u>Measure</u>	<u>Summary</u>
In-degree	For each node, sum of all incoming connections from other nodes to it
Out-degree	For each node, sum of all outgoing connections to other nodes from it
Total-degree	For each node, the combined measure of <i>in-</i> and <i>out-degree</i>
Betweenness	It measures the extent to which a node lies on paths between other nodes
Eigenvector	An extension to <i>degree centrality</i> (in-degree in a directed network). Takes into account the importance of neighbour nodes giving each node a score proportional to the sum of the scores of its neighbours
Closeness	Mean distance from a given node to the other nodes in the network

Table 44: Node-specific network measures

These measures are calculated for the different networks based on the three dependency corpora and their results compared in the next section.

7.3 Results

This section takes a look at the results of both general and node-specific measures. First, some general observations are presented, then, the SSD network is compared to the Temporal Dependencies network, and, afterwards, the SSD network is also compared to the Discourse Dependencies network. The different measures and their use in this evaluation are explained in Subsection 3.3.4. The following subsections show the obtained results of the evaluation and their interpretation.

7.3.1. General Observations in Previous Works

Several measures regarding complex networks are considered in this evaluation. Some measures give node-specific information, but other give information about the general

characteristics of the network. There is no specific work on Spanish dependency corpora available for a comparison of the results. Ferrer i Cancho et al. (2004) worked with dependency corpora in German, Czech and Romanian, and gives in this way a first idea of possible results. Therefore, the available results are compared in this subsection to the SSD model. Ferrer i Cancho et al. (2004) observe a small-world phenomenon in the syntactic dependency corpora. In case of Romanian, an average of 3.4 steps is measured for shortest paths between nodes in the network. SSD confirm this observation also for this Spanish corpus with a value of 3.1. The *transitivity* value of 0.02 in SSD is also within the range of the calculated values in Ferrer i Cancho et al. (2004). The measured *assortativity* value of - 0.22 shows disassortative mixing as expected since most networks, with the exception of social networks, show this characteristic (Ferrer i Cancho et al., 2004:3). The general characteristics of the network based on AnCora Surface Syntax Dependencies present comparable results to the observed data in Ferrer i Cancho et al. (2004).

Further general characteristics of the networks have not been calculated as the focus is on the optimisation process of the individual nodes considered relevant for a specific task.

7.3.2 Temporal Model

The different network measures have been compared between the network based on SSD (base model) and the one based on Temporal Dependencies (the optimised model for temporal parsing). The evaluation list, described in Subsection 7.2.4, has been used in order to obtain significant results. The presented results throughout this subsection refer to the sum of the obtained values of all nodes on the evaluation list.

The first measure of interest is in-degree, as it shows how many in-going connections a certain node receives within the network.

The following table gives an overview of the results⁸:

Measure	SSD sum	Temporal sum	Difference	Percentage
In-degree	69,602	84,984	(+ 15,382)	(+0.22 %)
Out-degree	35,877	44,657	(+ 8,780)	(+0.25 %)
Total-degree	105,479	129,641	(+ 24,162)	(+ 0.23 %)
Betweenness	305571455.8	367312517	(+ 61741061.2)***	(+ 0.20 %)
Eigenvector	8.45	9.93	(+ 1.48)***	(+ 0.18 %) ⁹
Closeness	944.65	947.39	(+2.74)***	(+ 0.03 %)

Table 45: Result comparison between SSD and Temporal Dependencies.

The incoming connections in a dependency structure show already at sentence level where the authority nodes are placed. The authorities take an important role in the parsing of the sentence. Therefore the in-degree is probably the most relevant measure among in, out- and total-degree in order to measure the importance of nodes in a language network. Figure 26 shows the incoming connections for the verb *vigilar* (‘to keep an eye on something’). Note that the used bracket indicates the direction of the connection. The increased importance in Temporal Dependencies can be seen by the higher number of lemmas which have *vigilar* as head, including a higher total frequency (from 19 to 34).

⁸ A dependent t-test (Student, 1908) for paired samples has been calculated for adequate measures and the results marked according to their p-value. If smaller than 0.05 with *, smaller than 0.01 with ** and smaller than 0.001 with ***.

⁹ Percentages have been calculated over the complete values, not over the indicated rounded numbers

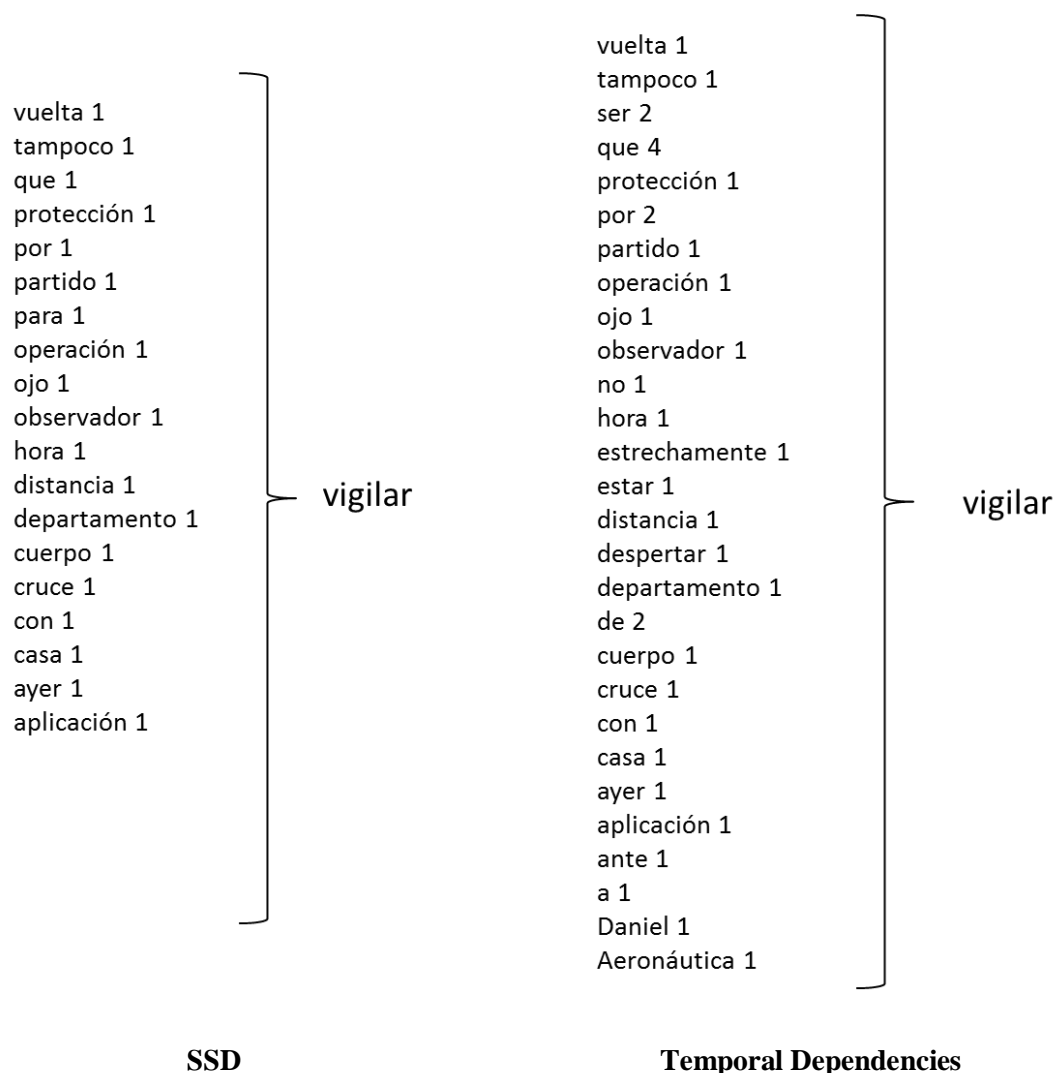


Figure 26: Incoming connections for *vigilar* (numbers show the frequency)

In general, the outgoing connections are equally important in a network analysis, but, since each node in a dependency structure will have exactly one head, the quantity of the out-degree shows only a change in lemma variation of the heads. This is normally due to an optimised version since important nodes, such as semantic information carrying verbs, were often connected to conjunctions and prepositions, which do not show a high variety, as they belong to a closed class. Nevertheless, the big change in importance comes here into play by a look at the quality of the heads, in this case “quality” refers to their relevance in the specific model. Semantic-information-carrying verbs have in both models a short path to the root in the created dependency structures

and it is expected that they connect to heads which are themselves semantic-information-carrying verbs. Figure 27 shows an example for the verb *vigilar*. While the sum of the frequencies is the same sum for the heads of the verbs in both models, the verb connects to a different type of heads. In Temporal Dependencies, the conjunction *que* ('that') and the connection to auxiliary verbs (e. g. *estar*, 'be' and *ser*, 'be') disappear from the list and *vigilar* connects now directly to two verbs which carry semantic information (*afirmar*, 'confirm' and *advertir*, 'warn'). Note that *vigilar* has been chosen to exemplify this treatment as the lemma itself does not show a high frequency in the corpus and allows a clear example. Other lemmas with higher frequencies in the corpus will benefit much more of the treatment.

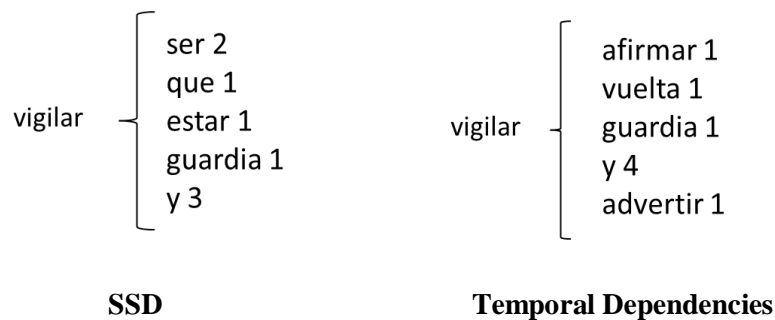


Figure 27: Outgoing connections of *vigilar*

The total-degree is the sum of in- and out-degree. An increase in its value after the optimisation process is therefore definitely a good sign. All three measures have seen an increase of at least a 22 % regarding their values in the base data. Especially the increase of a 22 % in the in-degree emphasises the gain of importance of the selected evaluation nodes in the temporal model. Their role as authorities has been enforced and this should have a positive influence on temporal parsing purposes over this data. Figure 28 shows a total degree example for *vigilar* in Temporal Dependencies.

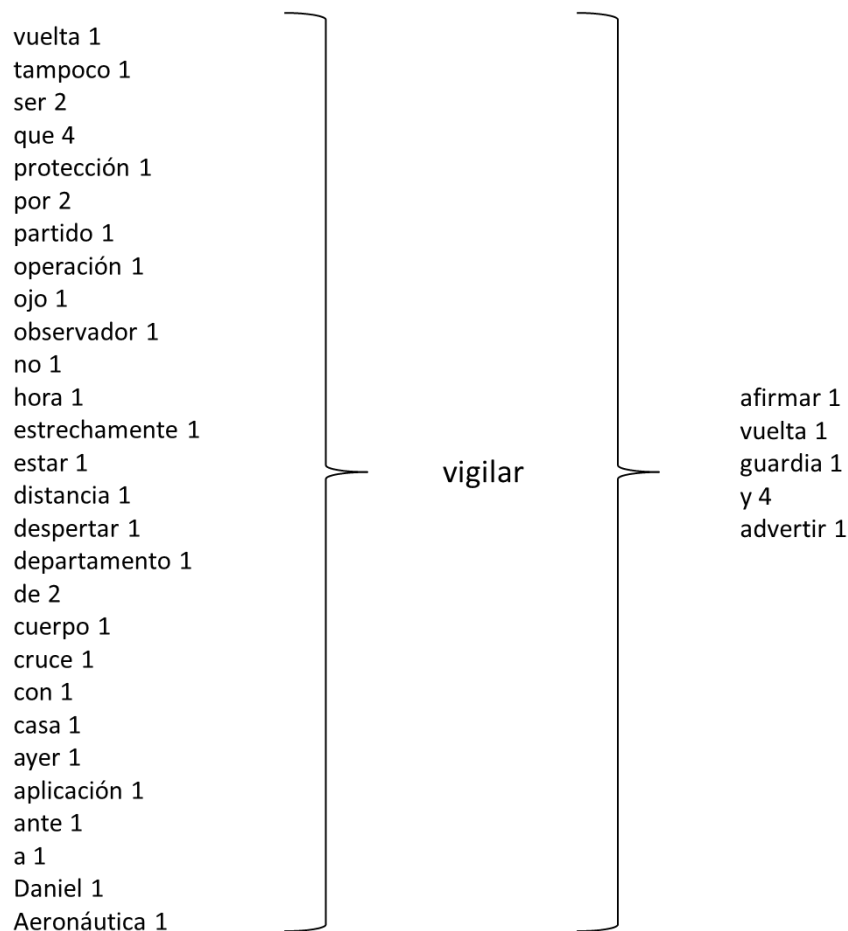


Figure 28: Total-Degree connections for *vigilar* (numbers indicate frequencies)

The observed increase of the incoming and outgoing edges has also a positive effect on another measure - *betweenness*. This measure basically counts how many times a certain node (a lemma) is on the way of the shortest path between any two nodes in the network. As the evaluated (and optimised) nodes are now more important in the network having more incoming and outgoing connections, their higher importance can also be observed in terms of *betweenness*. The increase of a 20 % (Table 45) shows that the optimised nodes lie now on paths with “more traffic” in the network which should facilitate parsing purposes. Figure 29 shows a simplified version of the graph for *vigilar*, which does not correspond to the full connections in the corpus. The example shows the shortest path for the noun *protección* (‘protection’) to the verb *ver* (‘see’). In

this case, both *vigilar* and *que* would get counted in terms of *betweenness* as they lay on the shortest path.

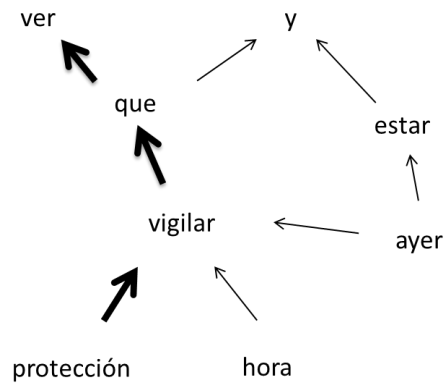


Figure 29: Betweenness graph example

Additionally, two other measures have been observed. Both make use of the concept of node neighbours in the network and include in different ways the importance of those. Figure 30 shows a simplified example for *vigilar*, in which its network neighbours are marked by a circle.

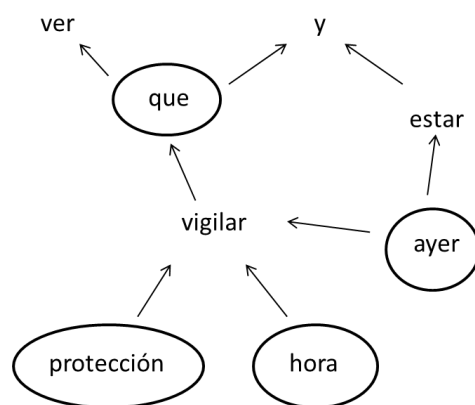


Figure 30: Node Neighbours in graph example

Eigenvector centrality is an extension to degree centrality as it takes into account also the importance of a node's neighbours. An increase in the present evaluation should therefore mean that an optimised and evaluated node has not only more connected

neighbour nodes in the network, but also neighbour nodes which are themselves more important. The measure is therefore relevant for the evaluation of the model optimisation as the evaluated nodes contain themselves relevant information for the specific parsing purposes and should connect as direct as possible to other relevant nodes. The increase of a 18 % in the evaluation confirms this optimisation process.

Finally, *closeness* has been calculated and shows a slight increase of 0.3 %. This is a result which makes its use for this evaluation doubtful since the change is really small and a negative value would show an optimisation. *Closeness* is a measure which is expected to show only small changes (Newman, 2010:182-183), so that this can still be a significant result. The problem with this measure is that it is a distance metric related to all nodes in the network. The task-based model does not optimise the connections for all nodes in the network but only for those which are considered as relevant for the task. Nodes that are not relevant for the task lose at the same time importance within the network. Therefore the measure does not reflect the conducted optimisation and the result should be similar to the general average path length in the network, which will be compared next.

Three general characteristics of the networks have also been compared. These measures are: *average path length*, *transitivity* and *assortativity*.

	SSD	Temporal Dependencies	Difference	Percentage
Average Path Length	3.10	3.11	(+ 0.01)	+ 0.01 %
Transitivity	0.015	0.018	(+ 0.003)	+ 0.20 %
Assortativity	- 0.22	- 0.20	(- 0.02)	+ 0.09 %

Table 46: General network characteristics for SSD and Temporal Dependencies

The *average path length* in the network has not improved, which is due to the fact that the optimisation of the model does not concern all nodes in the network. As most changes imply a reorganisation of the connections among nodes but not the disconnection of nodes in the dependency structure, the shortest path between relevant

nodes in the model will be optimised but it will also mean that at the same moment other nodes are not so well connected. This is perfectly fine for the optimisation idea of the task-based dependency customisation, but cannot be reflected in the *average path length* as general measure. On the other hand, the resulting changes in this value are so small that they do not have a deeper impact on the network.

The *transitivity* value for this language corpus network is first of all, as expected, not nearly as high as observed in social networks or other high clustering networks (Newman, 2010:200). But, the increase which can be found in the comparison of the SSD base model and the optimised temporal model is noticeable with around a 20 %. It is still not a highly clustered network in the temporal model, but the optimised nodes show a tendency to improve clustering within the network.

The measured *assortativity* value is negative for both models and therefore attributed to disassortative mixing. This was to be expected as most networks, with the exception of social networks, show this characteristic (Ferrer i Cancho et al., 2004:3). Nevertheless, the value increases in the temporal model by a 9 % and shows in this way a higher tendency to connect nodes with similar degrees.

7.3.3 Discourse Model

The same evaluation has also been done for the discourse model. Therefore, the previously explained evaluation list (Subsection 7.2.3) has been compared for the network based on the surface-syntax and the discourse model corpus.

The following table shows the obtained results:

Measure	SSD sum	Discourse sum	Difference	Percentage
In-degree	56,279	72,809	(+ 16,530)	(+ 0.30 %)
Out-degree	19,959	28,038	(+ 8,079)	(+ 0.41 %)
Total-degree	76,238	100,847	(+ 24,609)	(+ 0.32 %)
Betweenness	170972247.9	246497630.8	(+ 75525382.9)***	(+ 0.44 %)
Eigenvector	6.35	13.63	(+7.28)***	(+ 1.15 %)
Closeness	824.79	829.93	(+ 5.14)***	(+ 0.06 %)

Table 47: Result comparison between SSD and Discourse Dependencies

The observed results behave in a similar way to those observed for the network created by means of the temporal model.

In-, out- and total-degree show, as expected, a noticeable increase. The same can be said for *betweenness*. The reasons for this behaviour are the same as in the previously mentioned evaluation for the temporal model.

Eigenvector centrality sees a high boost and almost doubles its value. This increase is significantly higher than the one seen in the temporal model, this is possibly due to the node selection for the evaluation. While the evaluation list of the temporal model include also time expressions as nodes, the list for the discourse model only includes verbs. The latter are more likely to be well connected and to have *important* neighbours, as they connect directly to other verbs in both models.

Closeness again sees a slight increase, which underlines the hypothesis that it may not be an adequate measure to reflect the optimisation process of the task-based dependency models.

The general measures of the networks showed similar results as before compared to the temporal model.

	SSD	Discourse Dependencies	Difference	Percentage
Average Path Length	3.09	3.12	(+ 0.03)	(+ 0.01 %)
Transitivity	0.015	0.019	(+ 0.004)	(+ 0.27 %)
Assortativity	- 0.22	- 0.19	(- 0.03)	(+ 0.14 %)

Table 48: General network characteristics for SSD and Discourse Dependencies

Both *average path length* and *transitivity* values confirm the assumption made in the evaluation of the temporal model network. *Average path length* increases slightly, but not significantly, and *transitivity* gets a boost, but the network still maintains a non highly clustered nature. This is due to the fact that the optimisation process in this work does not improve the importance of all nodes in the network, but only of those which are considered relevant for the specific model. This optimisation normally implies a decline of importance for not relevant nodes. This fact has to be taken into account and shows why both measures do not reflect the implemented optimisation process.

The *assortativity* value, on the other hand, increases in a similar way for both task-optimised models. They show therefore both a higher tendency to cluster nodes with similar degrees.

7.4 Conclusion

The presented network creation and computation process in this chapter has been conducted mainly for two purposes: the interpretation of standard network analysis measures in a syntactic dependency network and the evaluation of the optimisation process of two previously created task-based dependency models by means of those measures.

Some measures are closely related to what was directly done in the optimisation process. The *in-degree* corresponds to the number of incoming nodes regarding a specific node in the network. This was exactly one of the basic ideas in the optimisation process, the direct connection of relevant nodes and the placement of those in *authority* positions which receive many incoming connections.

The *out-degree*, on the other hand, is limited in some way, as in a dependency structure each node only connects upwards to one single node. Nevertheless, this measure is interesting in order to see the variety of heads a specific node has. While in the SSD corpus verbs often connect to conjunctions or prepositions, they connect in the optimised models directly to other verbs which are themselves semantically significant. The obtained results correspond therefore to the expected optimisation.

The *total-degree* is the sum of both previous measures and therefore also relevant, even if the individual measures seem to give more detailed information.

Betweenness has shown to be another important measure for this evaluation. It counts basically how many times a specific node is on the way of the shortest path between any two other nodes in the network. This measure is somehow connected to the previously commented in- and out-degrees, since having many connections is useful to obtain high values for *betweenness*. It rewards well positioned nodes, which is an idea of the optimisation process. The obtained results have confirmed this assumption.

Eigenvector centrality is an extension to degree centrality which takes into account the importance of the neighbours of a specific node. The results have shown the expected improvement and shows further evidence for the optimisation process in the dependency model creation.

The only node-specific measure which has not proved its usefulness for the evaluation of the optimisation process has been *closeness*. It has shown a slight increase, which shows that the optimisation process has not improved the importance of all nodes in the network. This however was an expected result, since only nodes considered as relevant for a model were optimised, and this at the cost of the importance of not relevant nodes. Nevertheless, the latter do influence the *closeness* value. This makes this measure not adequate for the performed optimisation process.

Three measures of general network characteristics have been calculated and have shown results comparable to other language networks. *Average path length* and *transitivity* have not shown any improvements in the comparison between AnCora Surface Syntax Dependencies and the two task-based dependency corpora. They actually do not seem to

be a good option to measure the optimisation process conducted in the task-based dependency model creations. This is mainly due to the same reason that has been mentioned for *closeness*. The implemented optimisation improves the position of desired nodes in the network, but at cost of others nodes. The conducted optimisation is therefore not reflected in these two values. *Assortativity* on the other hand has improved along with the optimisation process in both cases, and shows an increase of the connection between nodes with similar degrees in the optimised networks.

Taking into account only the measures which reflect the implemented optimisation process between the base model and the two task-based models, the evaluation shows clearly proof for the desired optimisation of relevant nodes in the used dependency corpora. The measured optimisations are high enough to be considered as significant and should lead to a better performance in future usage of the data in specialised NLP applications.

8. Conclusion

The main goal of this dissertation was the task-based customisation of dependency structures. Two specific NLP tasks have been chosen to develop the customisation process: temporal and discourse parsing. A linguistic analysis of the treatment of time and discourse in Spanish has shown the information that is required for temporal and discourse parsing and how it can be informed in a dependency annotation. The observed linguistic criteria have been implemented in two task-oriented dependency models: Temporal and Discourse Dependencies.

The tailoring of the two different models has followed the same schema. They had in common that dependency structures could be modelled by the help of three basic means. Firstly, the head selection in dependency structures has made it possible to improve the positions of relevant nodes for the specific tasks. Secondly, the use of collapsed dependencies, and the implied disconnection of not relevant nodes, has further shortened the distances between relevant nodes. And thirdly, the tagset has been used to enhance the data with syntactic and task-specific information. The syntactic tags have been adapted to the chosen head selection criteria and reflect now the syntactic information that is required for the specific task. And the tags have been enriched in different ways with the readily available task-related information.

The task-specific information provided in Temporal Dependencies includes time expressions, which is essential for temporal parsing. The Discourse Dependencies model, on the other hand, has been enhanced by Discourse Dependencies Relations, an abstract version of ‘standard’ discourse relations, which will help discourse parsers with their work in the future.

The AnCora corpus (Taulé et al., 2008) has been used as starting point for this investigation, and as a result several new dependency annotation layers have been created: the AnCora Surface Syntax Dependencies, the AnCora Temporal Dependencies and the AnCora Discourse Dependencies. AnCora Surface Syntax Dependencies is a dependency annotation based on purely syntactic criteria, which serves as base data for the task-oriented model adaptation.

AnCora Temporal Dependencies and AnCora Discourse Dependencies are the result of adapting the base model to the two specific tasks chosen: temporal and discourse parsing respectively.

The three different versions of AnCora have shown that linguistic information can be encoded in dependency structures in different ways according to the further usage of the data. Not all NLP tools work with the same information, but depending on their task they have specific requirements on the information they use.

If they do not find this information explicitly in the input data, they will have to identify it themselves, and if the information is badly positioned within the dependency structures, it will be hard to find it. In both cases they are faced to a loss of performance.

The preparation of high quality linguistic annotations is at the base of the work with NLP tools (Pustejovsky and Stubbs, 2012). The scarcity of available data should not lead to the acceptance of the use of inadequate input data.

The work done within this project shows how the needed linguistic criteria and specific information can be implemented into a base dependency corpus in order to obtain an adapted version that can be used in a specific NLP task.

This adaptation can be conducted by automatic means, so that it can be inserted in an NLP pipeline. The dissertation's contributions can be seen in the following list. They include innovative theoretical approaches in the dependency area and the creation of various linguistic resources, which will be from now on available to the research community.

- **Contributions**
 - Theoretical investigation
 - Analysis of linguistic criteria needed for specific NLP tasks (focused on temporal and discourse parsing)
 - Task-based modelling of dependency structures by means of head selection, tagset and collapsed dependencies
 - Evaluation of a dependency structure optimisation process by means of network measures at general and node-specific level
 - Linguistic resources
 - AnCora Surface Syntax Dependencies
 - AnCora Temporal Dependencies
 - AnCora Discourse Dependencies
 - Code
 - Automatic Converter for AnCora constituents to Surface Syntax Dependencies
 - Automatic Adapter for Surface Syntax Dependencies to Temporal Dependencies
 - Automatic Adapter for Surface Syntax Dependencies to Discourse Dependencies
 - Publications
 - “From constituents to syntax-oriented dependencies” in Sociedad Española para el Procesamiento del Lenguaje Natural, Issue 52 (Kolz et al., 2014a)
 - “Multiword deconstruction in AnCora dependencies and final release data” (Kolz et al., 2014b)

The first practical step was the creation of a surface-syntax annotated corpus. It aimed at creating a purely syntactic dependency model, that is, a model in which only surface syntax facts were reflected. Most dependency models available have a tendency to adopt an (at least semi-) semantic approach to syntax dependencies. Of course the transition to a semantic focus makes sense since information extraction is one of the important tasks of NLP. But there are two problems with this. Firstly, the sole availability of semantic-oriented dependency corpora difficult the work when a purely syntax analysis is needed. And secondly, not all semantically focused tasks require the same kind of information in the data.

The created syntax-based dependency annotation proved to be a good base for the later adaptation to task-specific dependencies. The automatic conversion from constituents to dependencies has been performed and has delivered a new, purely syntactic, dependency annotation for Spanish. Two publications were written at that point (Kolz et al., 2014a and 2014b) and the first corpus was published, so that the research community could already access part of the contributions of this investigation. The AnCora Surface Syntax Dependencies corpus is available since the beginning of 2015 to the research community on the GLiCom¹⁰ website and on the official AnCora¹¹ website.

The creation of two task-specific models has shown how dependency annotations can vary in terms of head selection and encoded information according to their intended use. In this project, temporal and discourse parsing were selected as NLP tasks but other task-based models could be created. The automatic adaptation has shown good results when starting from a model based on surface syntactic information. Following this approach, automatic adapters could be implemented in NLP tools in the future, so that they can work with optimised input data and improve their results. The two created task-specific corpora can, from now on, be used by the appropriate NLP tools. They are already available to the linguistic research community for further investigation. The corpus AnCora benefits from the addition of the newly created resources. It was already a large project with a good amount of annotation levels and a considerable size in terms

¹⁰ http://www.upf.edu/glicom/recursos/corpus/ancora_ssd.html

¹¹ <http://clic.ub.edu/corpus/ancora-descarregues>

of tokens. The new resources offer a greater variety of dependency data, which will be useful for further linguistic projects. Both AnCora Temporal and Discourse Dependencies have been made available in December 2015 to the research community via the GLiCom and AnCora websites.

The final evaluation by means of a language network analysis has shed light on the characteristics the different dependency corpora, which are built on the same data, show when optimised according to specific linguistic criteria. The results have shown that the adaptation of the dependency models has resulted in an optimisation of the resulting networks. The general discussion of adequate network measures will help in future projects to compare dependency corpora by the use of network analysis, which is still at an initial stage when applied to language networks. While previous works on language networks (Ferrer i Cancho and Solé, 2001; Ferrer i Cancho et al., 2004) focused mainly on the general characteristics of those networks, the present project had the goal to look at specific nodes within the network in order to observe and compare their behaviour in three different models of data over the same corpus.

This investigation has selected several measures for this evaluation task, while ruling out one node-specific measure. There may be other candidates that can still show their adequacy in the future (e. g. Katz centrality), but this requires further research. The creation of several dependency models based on the same corpus has made it possible to investigate at a comparative level, which is useful for meaningful results.

As for the future work after the end of this dissertation several action lines can be envisaged. First, it will be interesting to see how much NLP tools can improve their performance by the use of task-adapted dependencies. This can be done by using created corpora as training data for machine learning algorithms. Both temporal and discourse parsing make use of machine learning approaches, which makes the use of the data feasible for their task.

Another action line in future work is the implementation of the automatic adaptation algorithm as a preprocessing function of the task-specific NLP tools. At the moment the created models have only been used to create a task-based dependency annotation for the AnCora corpus. Nevertheless, they are not AnCora bound. The only requirement

they have is to work over a base surface-syntax dependency annotation which satisfies the SSD model. The implementation of a complete preprocessing from plain text, over to a standard surface-syntax dependency annotation and the further adaptation to task-specific models is feasible by automatic means and left for future work.

Besides, other tasks within NLP could benefit as well of adapted dependency models. Such models can be created according to linguistic needs in the tasks. To this end a process can be followed that is similar to the one described throughout Chapters 4, 5 and 6 of this dissertation.

Network analysis is an area that is under heavy investigation in artificial intelligence, so that investigation in language networks can already rely on a good base. Nevertheless, the work using language data is still limited and it will be interesting to observe further results in future projects. This dissertation has taken a detailed look on specific nodes within language networks using already available measures to evaluate them.

In the future, it will be interesting to see results of comparable works but also to see if the way in which the dependency network representations are built influences the network analysis. For this dissertation it was decided to use an approach based on lemmas, as lemmas seemed to provide a better form of data representation than word forms, because of the generality that lemmas introduce. However it would be interesting to see whether the obtained results are maintained when using word forms to build the networks. Another possible research direction is to create the language network by means of more abstract representations, such as task-based categories. This was actually discussed within the work of this dissertation, but not approached due to time limits. In the network based on temporal dependencies, for example, the identified time expressions could be grouped together into a single “timex” node, or, in the discourse model, lexical discourse markers could be grouped into “discourse” nodes. Then, their position within the network could be analysed as a group and not by means of their lemma, as it has been done in the shown evaluation.

Bibliography

- Aparicio, J., Taulé, M. and Martí, M. A. (2008). AnCora-Verb: A Lexical Resource for the Semantic Annotation of Corpora. *Proceedings of 6th International Conference on Language Resources and Evaluation*. Marrakesh (Morocco).
- Arias, B., Bel, N., Fisas, B., Lorente, M., Marimon, M., Morell, C., Vázquez, S. and Vivaldi, J. (2014). The IULA Spanish LSP Treebank: building and browsing. *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC'14)*.
- Asher, N. (1993). *Reference to Abstract Objects in Discourse*. Kluwer, Dordrecht.
- Badia, T., Saurí, R. and Suñol, T. (2012). Guía de anotación de segmentos y relaciones discursivas. Unpublished.
- Becker, T., Joshi, A., and Rambow, O. (1991). Long distance scrambling and tree adjoining grammars. In *Fifth Conference of the European Chapter of the Association for Computational Linguistics (EACL '91)*, pp. 21–26. ACL.
- Borrega, O., Martí, M. A. and Taulé, M. (2007). What do we mean when we speak about Named Entities. In *Proceedings of Corpus Linguistics*.
- Bosque, I. and Demonte, V. (1999). *Gramática descriptiva de la lengua española*. Colección Nebrija y Bello, Espasa. Madrid.
- Brin, S. and Page, L. (1998). The anatomy of a large-scale hypertextual Web search engine. *CNIS*, 30(1-7):107-17.
- Caplan, D. (1987). *Neurolinguistics and linguistic aphasiology*. New York: Cambridge University Press.
- Carletta, J. (1996). Assessing agreement on classification task: the kappa statistic. *Computational Linguistics*, 22(2), pp. 249-254.
- Carlson, L., Marcu, D. and Okurowski, M. E., (2002). RST Discourse Treebank. Philadelphia: Linguistic Data Consortium.

- Čech, R., Mačutek, J., and Liu, H. (2016). Syntactic Complex Networks and Their Applications. In *Towards a Theoretical Framework for Analyzing Complex Linguistic Networks*, pp. 167-186. Springer Berlin Heidelberg.
- Chomsky, N. (1957). *Syntactic structures*. New York: Mouton.
- Chomsky, N. (1995). *The minimalist program*. Cambridge, MA: MIT Press.
- Choudhury, M. and Mukherjee, A. (2009). The structure and dynamics of linguistic networks. In N. Bellomo, N. Ganguly, A. Deutsch, and A. Mukherjee, editors. *Dynamics On and Of Complex Networks, Modeling and Simulation in Science, Engineering and Technology*, pp. 145-166. Birkhauser Boston.
- Choudhury, M., Chatterjee, D., and Mukherjee, A. (2010). Global topology of word co-occurrence networks: Beyond the two-regime power-law. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pp. 162-170, Association for Computational Linguistics.
- Civit, M. and Martí, M. A. (2004). Building Cast3LB: a Spanish Treebank. In *Research on Language & Computation (2004) 2*, pp. 549-574. Springer, Science & Business Media. Germany.
- Civit, M., Martí M. A., and Bui, N. (2006). Cat3LB and Cast3LB: From Constituents to Dependencies. In *Proceedings of the 5th International Conference on Natural Language Processing, FinTAL*, pp. 143-151, Turku, Finland. Springer Verlag LNAI 4139.
- Čmejrek, M., Cuřín, J. and Havelka, J. (2003). Czech-English Dependency-based Machine Translation. In *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics-Volume 1*, pp. 83-90. Association for Computational Linguistics.
- Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20, pp. 37-46.

- Collins, M. (1999). *Head-Driven Statistical Models for Natural Language Parsing*, PhD thesis, University of Pennsylvania.
- Comrie, B. (1976). *Aspect*. Cambridge University Press, Cambridge.
- Comrie, B. (1985). *Tense*. Cambridge University Press, Cambridge.
- Corominas-Murtra, B., Valverde, S. and Solé, R. V. (2010). Emergence of scale-free syntax networks. In: Nolfi, S. and Mirolli, M. (eds.) *Evolution of Communication and Language in Embodied Agents*, pp. 83-101. Springer, Heidelberg.
- Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20 (3), pp. 273-297.
- Csardi, G. and Nepusz T. (2006). The igraph software package for complex network research. *InterJournal, Complex Systems*, 1695(5), pp. 1-9.
- Culotta, A. and Sorensen, J. (2004). Dependency tree kernels for relation extraction, *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL)* (pp. 423-429), Barcelona, Spain.
- da Cunha, I., Torres-Moreno, J.-M. and Sierra, G. (2011). On the Development of the RST Spanish Treebank. In *Proceedings of the 5th Linguistic Annotation Workshop. 49th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 1-10. Portland, Oregon, USA: Association for Computational Linguistics.
- da Cunha, I., San Juan, E., Torres-Moreno, J.-M., Lloberes, M. and Castellón, I. (2012a). DiSeg 1.0: The First System for Spanish Discourse Segmentation. *Expert Systems with Applications (ESWA)* 39(2), pp. 1671-1678.
- da Cunha, I., San Juan, E., Torres-Moreno, J.-M., Cabré, T. and Sierra, G. (2012b). A Symbolic Approach for Automatic Detection of Nuclearity and Rhetorical Relations among Intra-sentence Discourse Segments in Spanish. In Gelbukh, A.

- (ed.). *Computational Linguistics and Intelligent Text Processing, Lecture Notes in Computer Science (LNCS) 7181*, pp. 462-474. Berlin: Springer.
- da Cunha, I., San Juan, E., Torres-Moreno, J.-M., Cabré, M. T., Sierra, G. (2012c). A Symbolic Approach for Automatic Detection of Nuclearity and Rhetorical Relations among Intra sentence Discourse Segments in Spanish. In: Gelbukh, A. (ed). *Computational Linguistics and Intelligent Text Processing. Lecture Notes in Computer Science (LNCS). CICLing 2012, Part I, 7181*, pp. 462-474. Springer Berlin Heidelberg.
- da Cunha, I. (2013). A Symbolic Corpus-based Approach to Detect and Solve the Ambiguity of Discourse Markers. *Research in Computing Science*, 70, pp. 93-104.
- Danlos, L., Gaiffe, B. and Roussarie, L. (2001). Document Structuring à la SDRT. In *Proceedings of the 8th European workshop on Natural Language Generation-Volume 8*, pp. 1-10. Association for Computational Linguistics.
- de Marneffe, M.-C. and Manning C. D. (2008). The Stanford typed dependencies representation. In *Coling 2008: Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation*, pp. 1-8. Association for Computational Linguistics.
- de Marneffe, M.-C. and Manning, C. D. (2012). Stanford typed dependencies manual. Technical report, Stanford University.
- de Marneffe, M.-C., Connor, M., Silveira, N., Bowman, S. R., Dozat, T. and Manning, C. D. (2013). More Constructions, more genres: Extending Stanford Dependencies. *DepLing 2013*, 187.
- de Marneffe, M.-C., Silveira, N., Dozat, T., Haverinen, K., Ginter, F., Nivre, J, and Manning, C. D. (2014). Universal Stanford dependencies: A cross-linguistic typology. In *Proc.of LREC'14*, Reykjavík, Iceland. European Language Resources Association (ELRA).

- Ding, Y. and Palmer, M. (2004). Synchronous dependency insertion grammars: A grammar formalism for syntax based statistical MT, *Proceedings of the COLING Workshop on Recent Advances in Dependency Grammar*, pp. 90-97, Switzerland.
- Duan, X., Zhao, J. and Xu, B. (2007). Probabilistic parsing action models for multilingual dependency parsing, *Proceedings of the CoNLL Shared Task of EMNLP-CoNLL 2007*, pp. 940-946, Prague, Czech Republic.
- duVerle, D, and Prendinger, H. (2009). A Novel Discourse Parser Based on Support Vector Machine Classification, *Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics*, Vol. 2, pp. 665–673.
- EAGLES (1996). Recommendations for the Morphosyntactic Annotation of Corpora. Available at: <http://www.ilc.cnr.it/EAGLES/annotate/annotate.html>. Last access: 12th January 2016.
- Eisner, J. M. (1996). Three new probabilistic models for dependency parsing: An exploration, *Proceedings of the 16th International Conference on Computational Linguistics (COLING)*, pp. 340-345, Copenhagen, Denmark.
- Everitt, B. (1998). *The Cambridge Dictionary of Statistics*. Cambridge, UK New York: Cambridge University Press.
- Ferrer i Cancho, R., and Solé, R. V. (2001). The small world of human language. *Proceedings of the Royal Society of London B: Biological Sciences* 268.1482, pp. 2261-2265.
- Ferrer i Cancho, R., Solé, R. V. and Köhler, R. (2004). Patterns in syntactic dependency networks. *Physical Review E* 69, 051915.
- Ferrer i Cancho, R. (2005). The structure of syntactic dependency networks: insights from recent advances in network theory. In: *The problems of quantitative linguistics*, Altmann, G., Levickij, V. and Perebyinis, V. (eds.). Chernivtsi: Ruta. pp. 60-75.

- Ferro, L., Mani, I., Sundheim, B., and Wilson, G. (2000). TIDES Temporal Annotation Guidelines, Draft v.1.0. Mitre technical report MTR00W0000094, MITRE.
- Ferro, L., Gerber, L., Mani, I., Sundheim, B., and Wilson, G. (2005). TIDES 2005 Standard for the Annotation of Temporal Expressions. Tech. rep., MITRE.
- Gaifman, H. (1965). Dependency systems and phrase structure systems, *Information and Control* 8, pp. 304-337.
- Gaizauskas, R. and Wilks, Y. (1998). Information extraction: Beyond document retrieval. *Journal of documentation*, 54(1), pp. 70-105.
- Galley, M. and McKeown, K. (2003). Improving word sense disambiguation in lexical chaining. In *IJCAI*, Vol. 3, pp. 1486-1488.
- Gerdes, K. (2006). Sur la non-équivalence des représentations syntaxiques: comment la représentation en X-barre nous amène au concept du mouvement. *Les Cahiers de Grammaire*. 30, pp. 175-192.
- Gerdes, K. and Kahane, S. (2007). Phrasing It Differently. Wanner, L. (ed.). *Selected lexical and grammatical issues in the Meaning-Text Theory*, pp. 297-335. Amsterdam; Philadelphia: Benjamins.
- Grishman, R., and Sundheim, B. (1996). Message understanding conference-6: A brief history. In *COLING*, Vol. 96, pp. 466-471.
- Grosz, B. J. and Sidner, C. L. (1986). Attention, intention, and the structure of discourse. *Computational Linguistics* 12(3), pp. 175-204.
- Haghighi, A., Ng, A., Manning, C. (2005). Robust textual inference via graph matching. *Proceedings of the Human Language Technology Conference and the Conference on Empiric Methods in Natural Language Processings (HLT/EMNLP)*, pp. 387-394, Vancouver, Canada.

- Hajic, J. (1999). Building a syntactically annotated corpus: the Prague Dependency Treebank. *Issues in Valency and Meaning. Studies in honour of Jarmila Panevova.*
- Hajič, J., Hajicová, E., Panevová, J., Sgall, P., Bojar, O., Cinková, S., and Žabokrtský, Z. (2012). Announcing Prague Czech-English Dependency Treebank 2.0. In *Proceedings of the 8th International Conference on Language Resources and Evaluation*, pp. 3153-3160, Istanbul, Turkey.
- Harris, Z. S. (1968). *Mathematical Structures of Language*. Wiley.
- Hays, D. (1964). Dependency theory: A formalism and some observations, *Language* 40, pp. 511-525.
- Hellwig, P. (1986). Dependency Unification Grammar, *Proceedings of the 11th International Conference on Computational Linguistics (COLING)*, pp. 195-198, Bonn, Germany.
- Hellwig, P. (2003). Dependency Unification Grammar. In Agel, V, Eichinger, L. M., Eroms, H.-W., Hellwig, P., Heringer, H. J. and Lobin, H. (eds), *Dependency and Valency*, Walter de Gruyter, pp. 593-635.
- Hernault, H., Prendinger, H., duVerle, D. and Ishizuka, M. HILDA. (2010). A discourse parser using Support Vector Machine classification, *Dialogue and Discourse*. Vol. 1, No. 3, pp. 1-33.
- Hobbs, J. R. (1985). On the Coherence and Structure of Discourse. CSLI, pp. 85-37, Center for the Study of Language and Information, Stanford University.
- Hudson, R. A. (1984). *Word Grammar*, Blackwell.
- Hudson, R. A. (1990). *English Word Grammar*, Blackwell.
- Iruskieta M., Diaz de Ilarraza A., Labaka G., and Lersundi M. (2015). The Detection of Central Units in Basque scientific abstracts. *5th Workshop "RST and Discourse*

Studies", in *Actas del XXXI Congreso de la Sociedad Española del Procesamiento del Lenguaje Natural (SEPLN 2015)*, Alicante (Spain).

Iruskieta M. and Zapirain B. (2015). EusEduSeg: a Dependency-Based EDU Segmentation for Basque. In *Actas del XXXI Congreso de la Sociedad Española del Procesamiento del Lenguaje Natural (SEPLN 2015)*, pp. 41-48. Alicante (Spain).

Iruskieta, M., da Cunha, I. and Taboada, M. (2015). Principles of a qualitative method for rhetorical analysis evaluation: A contrastive analysis English-Spanish-Basque. *Language Resources and Evaluation* 49 (2): pp. 263-309.

Jeong, H., Mason, S. P., Barabási, A.-L. and Oltvai, Z. N. (2002). Lethality and centrality in protein networks. *Nature* 411, pp. 41-42.

Johansson, R. and Nugues, P. (2007). Extended constituent-to-dependency conversion for English, *Proceedings of NODALIDA 2007*, pp. 105-112, Tartu, Estonia.

Joty, S., Carenini, G., Ng, R., and Mehdad, Y. (2013). Combining intra- and multisentential rhetorical parsing for document-level discourse analysis. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL 2013)*, pp. 486-496, Sofia, Bulgaria.

Kamp, H. (1981). A Theory of Truth and Semantic Representation. In Groenendijk, Jeroen, J., Theo M.V. and Stokhof, M. (eds.) *Formal Methods in the Study of Language, Part 1*. Mathematical Centre Tracts, pp. 277–322.

Kamp, H. and Reyle, U. (1993). *From Discourse to Logic*. Kluwer Academic Publishers, Dordrecht.

Kapatsinski, V. (2006). Sound similarity relations in the mental lexicon: Modeling the lexicon as a complex network. *Speech Research Lab Progress Report, Indiana University*.

Katz, L. (1953). A new status index derived from sociometric analysis. *Psychometrika*, 18(1), pp. 39-43.

- Kazakov, D. and Sedding, J. (2004). Wordnet-Based Text Document Clustering. Third Workshop on Robust Methods in Analysis of Natural Language Data (ROMAND), pp. 104-113.
- Klein, D. (2015). *The Unsupervised Learning of Natural Language Structure*, PhD thesis, Stanford University.
- Kolomiyets, O., and Moens, M. (2013). KUL: A data-driven approach to temporal parsing of documents. *Proceedings of the second joint conference on lexical and computational semantics (* SEM)*, 2, pp. 83-87.
- Kolz, B. (2012). ENTImex – Extractor and Normalizer of Time Expressions: A rule-based approach to temporal expression extraction and normalization for Spanish. Unpublished.
- Kolz, B., Badia, T. and Saurí, R. (2014a). From constituents to syntax-oriented dependencies. *Procesamiento del Lenguaje Natural*, 52, pp. 53-60.
- Kolz, B., Badia, T. and Saurí, R. (2014b). Multiword deconstruction in AnCora dependencies and final release data. Available at: https://www.upf.edu/glicom/_pdf/Technical_Report_GLiCom_2014-1-AnCora_SSD.pdf. Last access: 18th January 2016.
- Kong, L., Rush, A. M., and Smith, N.A. (2015). Transforming dependencies into phrase structures. In *Proc. of the Conference of the North American Chapter of the Association for Computational Linguistics*.
- Kübler, S., McDonald, R. and Nivre, J. (2009). *Dependency Parsing*. Synthesis Lectures on HLT. Morgan and Claypool Publishers.
- Kudo, T. and Matsumoto, Y. (2002). Japanese dependency structure analysis based on support vector machines, *Proceedings of the 6th Workshop on Computational Language Learning (CoNLL)*, pp. 63-69, Taipei, Taiwan.
- Laferty, J. D., McCallum, A. and Pereira, F. C. N. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings*

of the 18th ICML, pp. 282-289. Morgan Kaufmann.

Landis, J. R. and Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics* 33 (1), pp. 159–174.

Lascarides, A. and Asher, N. (1993). Temporal interpretation, discourse relations and commonsense entailment. *Linguistics and philosophy*, 16(5), pp. 437-493.

Lascarides, A. and Asher, N. (2007). Segmented discourse representation theory: Dynamic semantics with discourse structure. In Bunt, H. and Muskens, R. eds, *Computing Meaning: Volume 3*, Kluwer Academic Publishers, Dordrecht.

LeThanh, H., Abeyasinghe, G., and Huyck, C. (2004). Generating Discourse Structures for Written Texts. In *Proceedings of the 20th international conference on Computational Linguistics, COLING '04*, Geneva, Switzerland. Association for Computational Linguistics.

Levy, O. and Goldberg, Y. (2014). Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Baltimore, Maryland, USA, June. Association for Computational Linguistics.

Lin, Z., Kan, M.-Y., and Ng, H. T. (2009). Recognizing implicit discourse relations in the Penn Discourse Treebank. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, Volume 1, EMNLP '09*, pp. 343-351.

Lin, Z., Ng, H. T., and Kan, M.-Y. (2010). A PDTB-styled end-to-end discourse parser. Technical report, School of Computing, National University of Singapore.

Liu, Y., Wei, F., Li, S., Ji, H., Zhou, M., and Wang, H. (2015). A Dependency-Based Neural Network for Relation Classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pp. 285-290, Beijing, China. Association for Computational Linguistics.

- Llorens, H., Saquete, E., and Navarro, B. (2010). TipSem (English and Spanish): Evaluating CRFs and Semantic Roles in TempEval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pp. 284-291.
- Llorens, H. (2011). *A Semantic Approach to Temporal Information Processing*. PhD thesis. University of Alacant.
- Llorens, H., Saquete, E., Navarro-Colorado, B. and Gaizauskas, R. (2011). Time-Surfer: Time-based Graphical Access to Document Content. In Clough, P., Foley, C., Gurrin, C., Jones, G., Kraaij, W., Lee, H. and Mudoch, V. (eds), *Advances in Information Retrieval, vol. 6611 of Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 767-771.
- Louis, A., Joshi, A., and Nenkova, A. (2010). Discourse Indicators for Content Selection in Summarization. In *Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, SIGDIAL '10*, pp. 147-156, Tokyo, Japan. Association for Computational Linguistics.
- Ma, M., Huang, L., Zhou, B., and Xiang, B. (2015). Dependency-based Convolutional Neural Networks for Sentence Embedding. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pp. 174-179, Beijing, China. Association for Computational Linguistics.
- Magerman, D. (1994). *Natural language parsing as statistical pattern recognition*. Ph.D. thesis, Stanford University.
- Mani, I., Pustejovsky, J. and Gaizauskas, R. (Eds.) (2005). *The Language of Time*, Oxford University Press, Oxford.
- Mann, W. C. and Thompson, S. A. (1986). Relational propositions in discourse. *Discourse Processes* 9 (1), pp. 57-90.

- Mann, W. C. and Thompson, S. A. (1988). Rhetorical structure theory: Toward a functional theory of text organization. *Text-Interdisciplinary Journal for the Study of Discourse*, 8(3), pp. 243-281.
- Marcu, D. (1997). The Rhetorical Parsing of Natural Language Texts. *The Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics, (ACL'97/EACL'97)*, pp. 96-103, Madrid, Spain.
- Marcu, D. (2000a). The Rhetorical Parsing of Unrestricted Texts: A Surface-based Approach. *Computational Linguistics*, 26, pp. 395–448.
- Marcu, D. (2000b). *The Theory and Practice of Discourse Parsing and Summarization*. MIT Press, Cambridge, MA, USA.
- Martí, M. A. and Taulé, M. (2007). CESS-ECE: Corpus anotados del español y catalán. *Arena Romanística, 1*. A new Nordic journal of Romance studies. Bergen, Norway.
- Maruyama, H. (1990). Structural disambiguation with constraint propagation, *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 31-38, Pittsburgh, PA.
- Maziero, E., Pardo, T. A. S., da Cunha, I., Torres-Moreno, J-M., SanJuan, E. (2011). DiZer 2.0 - An Adaptable On-line Discourse Parser. In: *Proceedings of the III RST Meeting (8th Brazilian Symposium in Information and Human Language Technology)*, pp. 50-57.
- Mel'čuk, I. (1981). Meaning-text models: A recent trend in Soviet linguistics. *Annual Review of Anthropology*, pp. 27-62.
- Mel'čuk I. and Pertsov N. (1987). *Surface syntax of English: A formal model within the meaning-text framework* (Vol. 13). John Benjamins Publishing.
- Mel'čuk, I. (1988). *Dependency Syntax: Theory and Practice*, State University of New York Press.

- Mille, S., Burga, A., Vidal, V. and Wanner, L. 2009. Towards a Rich Dependency Annotation of Spanish Corpora. In *Proceedings of SELPN'09*, pp. 325-333, San Sebastian, Spain.
- Miller, G. A. (1995). WordNet: A Lexical Database for English. *Communications of the ACM Vol. 38*, No. 11, pp. 39-41.
- Miller, G. A. and Gildea, P. M. (1987). How children learn words. *Scientific American*, 257(3), pp. 86-91.
- Mitchell, T. M., Shinkareva, S. V., Carlson, A., Chang, K. M., Malave, V. L., Mason, R. A., and Just, M. A. (2008). Predicting human brain activity associated with the meanings of nouns. *Science*, 320(5880), pp. 1191-1195.
- Moreno, A., Grishman, R., López, Sánchez, F., and Sekine, S. (2000). Treebank of Spanish and its Application to Parsing. In *Proceedings of LREC-2000*, Athens, Greece.
- Motter, A.E., de Moura, A.P.S., Lai, Y.C., Dasgupta, P. (2002). Topology of the conceptual network of language. *Physical Review E*, 65(6), 065102.
- Mozgovoy, M. (2011). Dependency-Based Rules for Grammar Checking with Language Tool. *Proceedings of the Federated Conference on Computer Science and Information Systems*, pp. 209-212.
- Mukherjee, A., Choudhury, M., Basu, A., and Ganguly, N. (2007). Self-organization of sound inventories: Analysis and synthesis of the occurrence and co-occurrence networks of consonants. *Journal of Quantitative Linguistics*, 16(2), pp. 157-184.
- Negri, M. and Marseglia, L. (2004). Recognition and Normalization of Time Expressions: ITC-irst at TERN 2004. Trento.
- Newman, M. E. J. (2010). *Networks: An Introduction*. Oxford University Press, Oxford.
- Nivre, J, Hall, J. and Nilsson, J. (2006). MaltParser: A Data-Driven Parser-Generator for Dependency Parsing. In *Proceedings of LREC*, pp. 2216-2219.

- Page, L., Brin, S., Motwani, R. and Winograd, T. (1998). The PageRank citation ranking: Bringing order to the Web.
- Palmer, F. R. (2001). *Mood and Modality*. 2nd edition. Cambridge University Press, Cambridge.
- Pardo, T., and Nunes, M. (2008). On the development and evaluation of a Brazilian Portuguese discourse parser. *Journal of Theoretical and Applied Computing*, 15(2), pp. 43-64.
- Patwardhan, S., and Pedersen, T. (2006). Using wordnet-based context vectors to estimate the semantic relatedness of concepts. In *Proceedings of the Workshop on Making Sense of Sense at the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL-2006)*, pp. 1–8.
- Pitler, E., Raghupathy, M., Mehta, H., Nenkova, A., Lee, A., and Joshi, A. K. (2008). Easily identifiable discourse relations.
- Pitler, E., and Nenkova, A. (2009). Using Syntax to Disambiguate Explicit Discourse Connectives in Text. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pp. 13-16.
- Prasad, R., Joshi, A., Dinesh, N., Lee, A., Miltsakaki, E., and Webber, B. (2005). The Penn Discourse TreeBank as a Resource for Natural Language Generation. In *Proceedings of the Corpus Linguistics Workshop on Using Corpora for Natural Language Generation*, pp. 25-32, Birmingham, U.K.
- Prasad, R., Dinesh, N., Lee, A., Miltsakaki, E., Robaldo, L., Joshi, A., and Webber, B. (2008). The Penn Discourse Treebank 2.0. In *Proceedings of LREC'08*.
- Pustejovsky, J., Castano, J. M., Ingria, R., Sauri, R., Gaizauskas, R. J., Setzer, A., and Radev, D. R. (2003). TimeML: Robust specification of event and temporal expressions in text. *New directions in question answering*, 3, pp. 28-34.

- Pustejovsky, J., Knippen, B., Littman, J., and Saurí, R. (2005). Temporal and event information in natural language text. *Language Resources and Evaluation*, 39(2), pp. 123–164.
- Pustejovsky, J. and Stubbs, A. (2012). *Natural Language Annotation and Machine Learning*. O'Reilly Publishers.
- Pyysalo S., Ginter, F., Heimonen, J., Björne, J., Boberg, J., Järvinen, J., and Salakoski, T. (2007). BioInfer: A corpus for information extraction in the biomedical domain. *BMC Bioinformatics*, 8(50).
- Quirk, C., Menezes, A., and Cherry, C. (2005). Dependency treelet translation: syntactically informed phrasal SMT, *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, Ann Arbor, MI, pp. 271-279.
- Rambow, O. (2010). The simple truth about dependency and phrase structure representations: An opinion piece. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 337-340, Los Angeles, CA.
- Sagae, K. (2009). Analysis of discourse structure with syntactic dependencies and data-driven shift-reduce parsing. In *Proceedings of the 11th International Conference on Parsing Technologies (IWPT'09)*, pp. 81-84, Paris, France, Association for Computational Linguistics.
- Saquete, E., Muñoz, R., and Martínez-Barco, P. (2006). Event ordering using TERSEO system. *Data Knowledge Engineering*, 58 (1), pp. 70–89.
- Saurí, R., Littman, J., Knippen, B., Gaizauskas, R., Setzer, A., and Pustejovsky, J. (2005). TimeML Annotation Guidelines.
- Saurí, R. (2008). *A Factuality Profiler for Events in Text*. PhD Dissertation. Brandeis University.

- Saurí, R., Saquete, E., and Pustejovsky, J. (2010). Annotating Time Expressions in Spanish. *TimeML Annotation Guidelines (Version TempEval-2010)*. Barcelona Media. Technical Report BM 2010-02.
- Schilder, F., Katz, G., and Pustejovsky, J. (2007). *Annotating, Extracting and Reasoning about Time and Events*. Springer Verlag, Berlin.
- Schwartz, R., Abend, O., and Rappoport, A. (2012). Learnability-based syntactic annotation design. In *COLING 24*, pp. 2405-2422.
- Setzer, A., and Gaizauskas, R. (2000). Annotating Events and Temporal Information in Newswire Texts. In *LREC 2000*, pp. 1287-1294.
- Setzer, A. (2001). *Temporal information in newswire articles: An annotation scheme and corpus study*. Ph.D. thesis, University of Sheffield.
- Sigman, M. and Cecchi, G. (2002). Global organization of the Wordnet lexicon. *Proceedings of the National Academy of Sciences*, 99(3), pp. 1742-1747.
- Silveira, N., Dozat, T., de Marneffe, M.-C., Bowman, S., Connor, M., Bauer, J., and Manning, C. D. (2014). A gold standard dependency corpus for English. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*.
- Silveira, N. and Manning, C. (2015). Does universal dependencies need a parsing representation? An investigation of English, *Depling 2015*, p. 310.
- Snow, R., Jurafsky, D. and Ng, A. Y. (2005). Learning syntactic patterns for automatic hypernym discovery, *Advances in Neural Information Processing Systems (NIPS)*, Vancouver, Canada.
- Solan, Z., Horn, D., Ruppin, E., and Edelman, S. (2005). Unsupervised learning of natural languages. *Proceedings of National Academy of Sciences*, 102(33), pp. 11629-11634.
- Solé, R. V. (2005). Syntax for free? *Nature* 434, 289.

- Solé, R. V., Corominas-Murtra, B., Valverde, S., and Steels, L. (2010). Language networks: their structure, function, and evolution. *Complexity*, 15(6), pp. 20-26.
- Starosta, S. (1988). *The Case for Lexicase: An Outline of Lexicase Grammatical Theory*, Pinter Publishers.
- Steyvers, M., and Tenenbaum, J. B. (2005). The Large-scale structure of semantic networks: Statistical analyses and a model of semantic growth. *Cognitive science*, 29(1), pp. 41-78.
- Strötgen, J. and Gertz, M. (2010). HeidelTime: High quality rule-based extraction and normalization of temporal expressions. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pp. 321-324.
- Student. (1908). *The Probable Error of a Mean*. In: *Biometrika*. 6, pp. 1-25.
- Taboada, M. and Mann, W. C. (2006). Applications of Rhetorical Structure Theory. *Discourse Studies* 8 (4), pp. 567-588.
- Taboada, M. and Mann, W. C.. (2006). Rhetorical Structure Theory: Looking Back and Moving Ahead. *Discourse Studies* 8(3), pp. 423-459.
- Taboada, M. and Habel, C. (2013). Rhetorical relations in multimodal documents. *Discourse Studies* 15 (1), pp. 59-85.
- Tamariz, M. (2005). *Exploring the Adaptive Structure of the Mental Lexicon*. PhD thesis, Department of theoretical and applied linguistics, University of Edinburgh.
- Taylor, A., Marcus, M., and Santorini, B. (2000). The Penn treebank: an overview. In *Treebanks*, pp. 5-22. Springer Netherlands.
- Taulé, M., Martí, M. A., and Recasens, M. (2008). AnCora: Multilevel Annotated Corpora for Catalan and Spanish. *Proceedings of 6th International Conference on Language Resources and Evaluation*, pp. 96-101, Marrakesh, Morocco.
- Tesnière, L. (1959). *Éléments de syntaxe structurale*, Klincksieck, Paris.

- Tofiloski, M., Brooke, J., and Taboada, M. (2009). A syntactic and lexical-based discourse segmenter. In *Proceedings of the 47th annual meeting of the association for computational linguistics*, Singapore, pp. 77–80.
- Uriagereka, J. (1998). *Rhyme and Reason. An introduction to minimalist syntax*. Cambridge, MA: MIT Press.
- UzZaman, N., Llorens, H., Derczynski, L., Allen, J., Verhagen, M., and Pustejovsky, J. (2013). Semeval-2013 task 1: Tempeval-3: Evaluating time expressions, events, and temporal relations. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pp. 1-9, Atlanta, Georgia, USA, June. Association for Computational Linguistics.
- Van der Beek, L., Bouma, G., Malouf, R. and van Noord, G. (2002). The Alpino Dependency Treebank. In *Computational Linguistics in the Netherlands CLIN 2001*.
- Vera-Morales, J. (2004). *Spanische Grammatik*. 4. Auflage. Oldenbourg, Munich.
- Verberne, S., Boves, L., Oostdijk, N. and Coppen, P. (2007). Evaluating Discourse-based Answer Extraction for Why-question Answering. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 735-736, Amsterdam, The Netherlands. ACM.
- Vitevitch, M. S. (2004). *Phonological Neighbors in a Small World*. Ms. University of Kansas.
- Wang, M., Smith, N. A. and Mitamura, T. (2007). What is the Jeopardy Model? A quasi-synchronous grammar for QA, *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Processing (EMNLP-CoNLL)*, pp. 22-32, Prague, Czech Republic.
- Wang, R. and Zhang, Y. (2010). Hybrid Constituent and Dependency Parsing with Tsinghua Chinese Treebank. In *Proceedings of LREC-2010*, Valetta, Malta.

- Webber, B. (2004). D-LTAG: Extending lexicalized TAG to discourse. *Cognitive Science*, 28(5), pp. 751-779.
- Webber, B. L., Knott, A., Stone, M. and Joshi, A. K. (1999). Discourse relations: A structural and presuppositional account using lexicalized TAG. Paper presented at the 37th annual meeting of the Association for Computational Linguistics (ACL-99), College Park, MD.
- Wei Feng, V. and Hirst, G. (2012). Text-level discourse parsing with rich linguistic features. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-2012)*, pp. 60-68, Jeju, Korea.
- Wei Feng, V. and Hirst, G. (2014). A Linear-Time Bottom-Up Discourse Parser with Constraints and Post-Editing. In *Proceedings of the 52th Annual Meeting of the Association for Computational Linguistics (ACL-2014)*, pp. 511-521, Baltimore, USA.
- Wellner, B. (2009). *Sequence Models and Ranking Methods for Discourse Parsing*. PhD Dissertation, Brandeis University.
- Wilson, G., Mani, I., Sundheim, B., and Ferro, L. (2001). A multilingual approach to annotating and extracting temporal information. In *Proceedings of the workshop on Temporal and Spatial information processing*, pp. 81-87, NJ, USA. ACL.
- Wolf, F. and Gibson, E. (2006). *Coherence in Natural Language*. MIT Press. Massachusetts Institute of Technology.
- Yamada, H. and Matsumoto, Y. (2003). Statistical dependency analysis with support vector machines. In *Proceedings of 8th International Workshop on Parsing Technologies*, pp. 195-206.
- Yoshida, Y., Suzuki, J., Hirao, T., and Nagata, M. (2014). Dependency-based Discourse Parser for Single-Document Summarization. In *Proceedings of Empirical Methods for Natural Language Processing (EMNLP)*.

Zipf, G. K. (1935). *The Psychobiology of Language*. Houghton-Mifflin, Boston, USA.

Zipf, G. K. (1949). Human behavior and the principle of least effort.